

FIG. 1

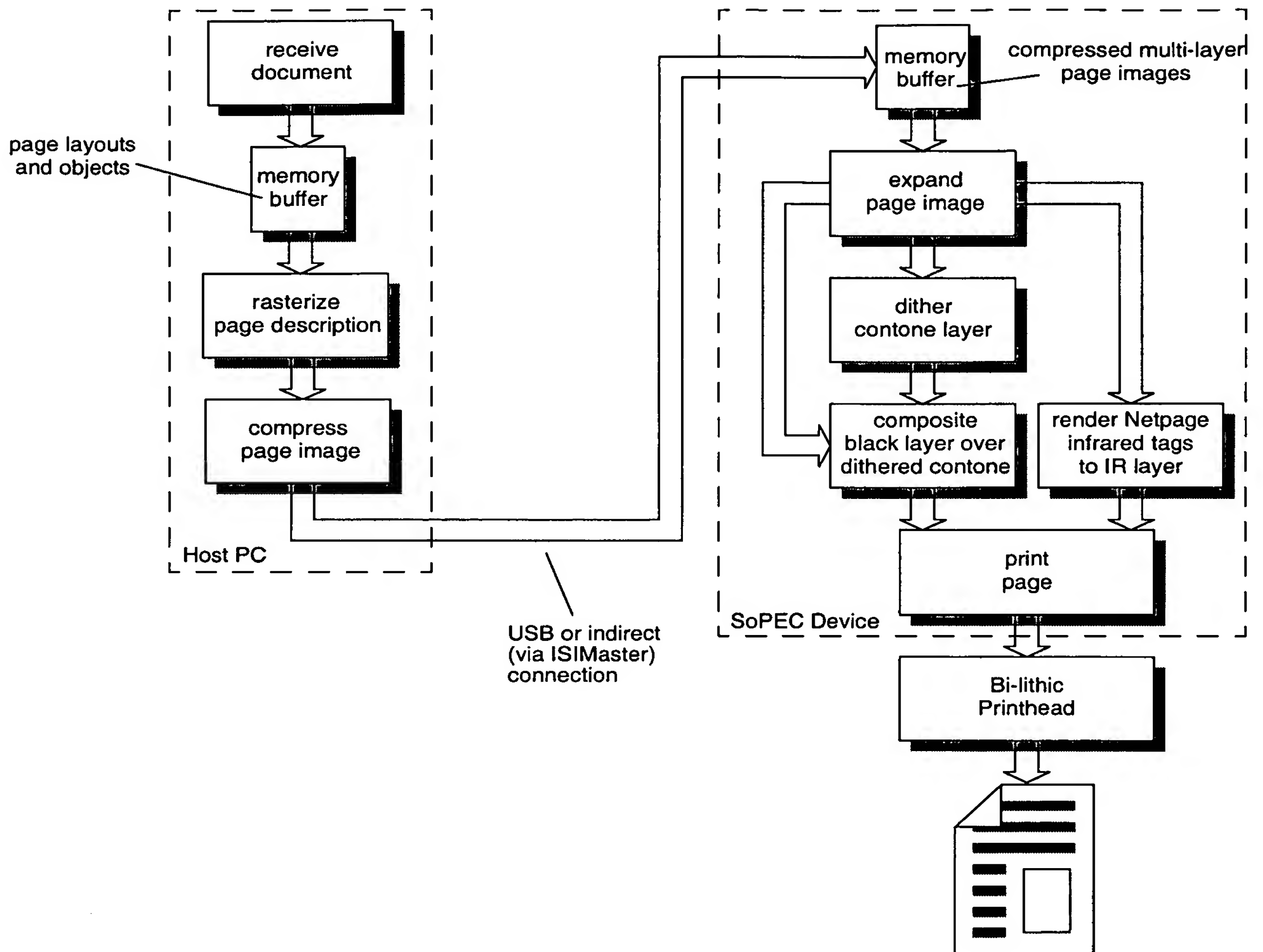


FIG. 2

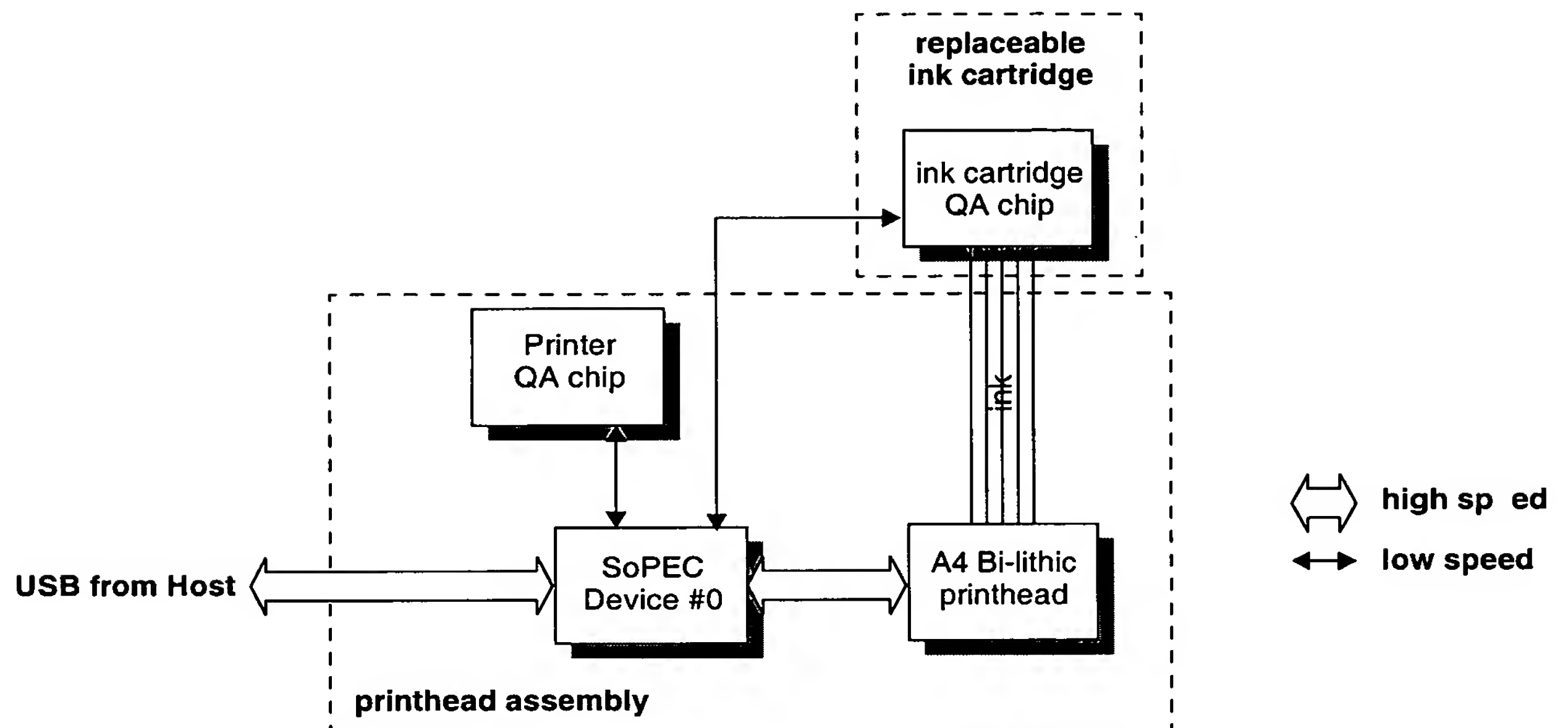


FIG. 3

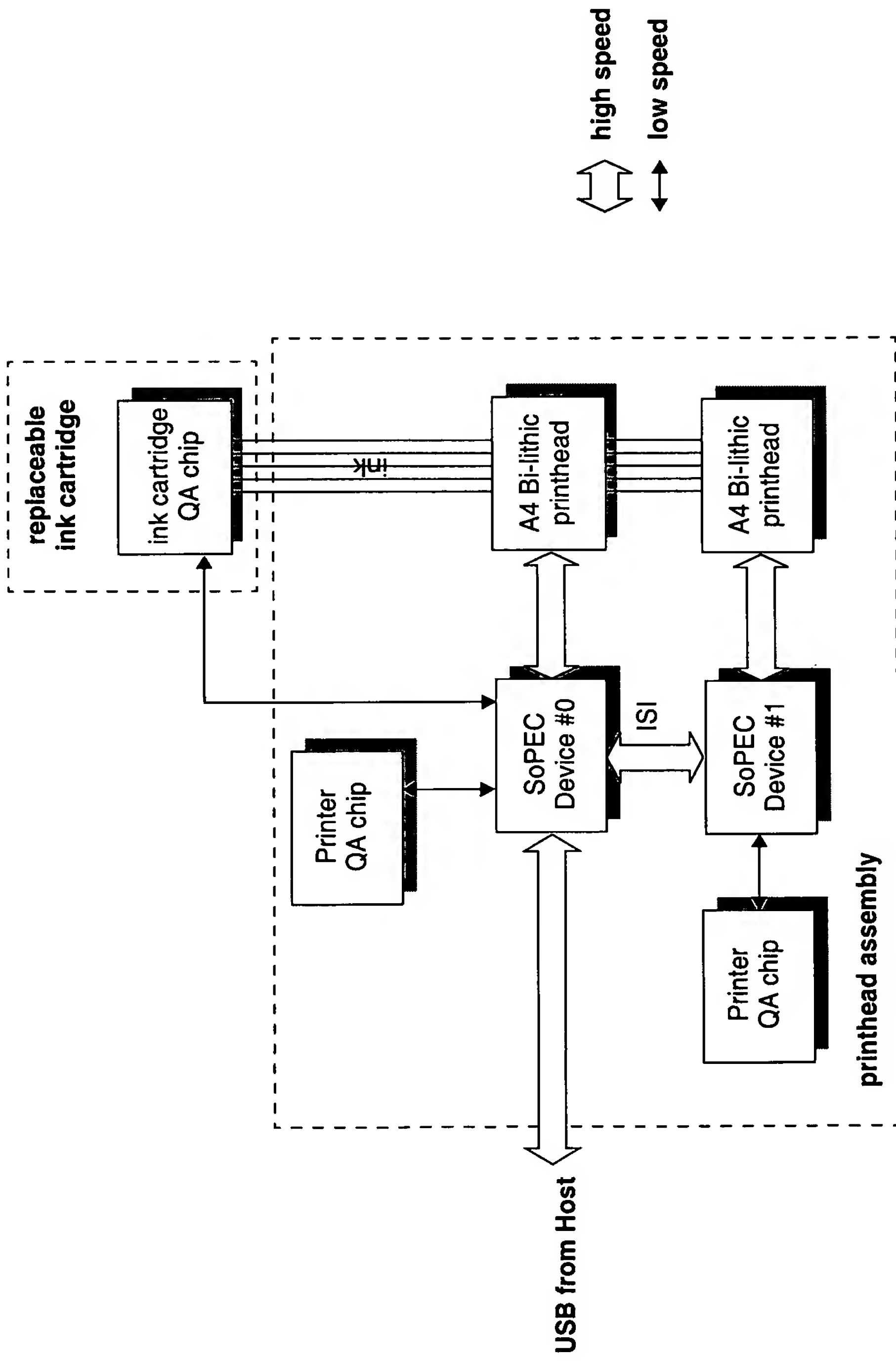


FIG. 4

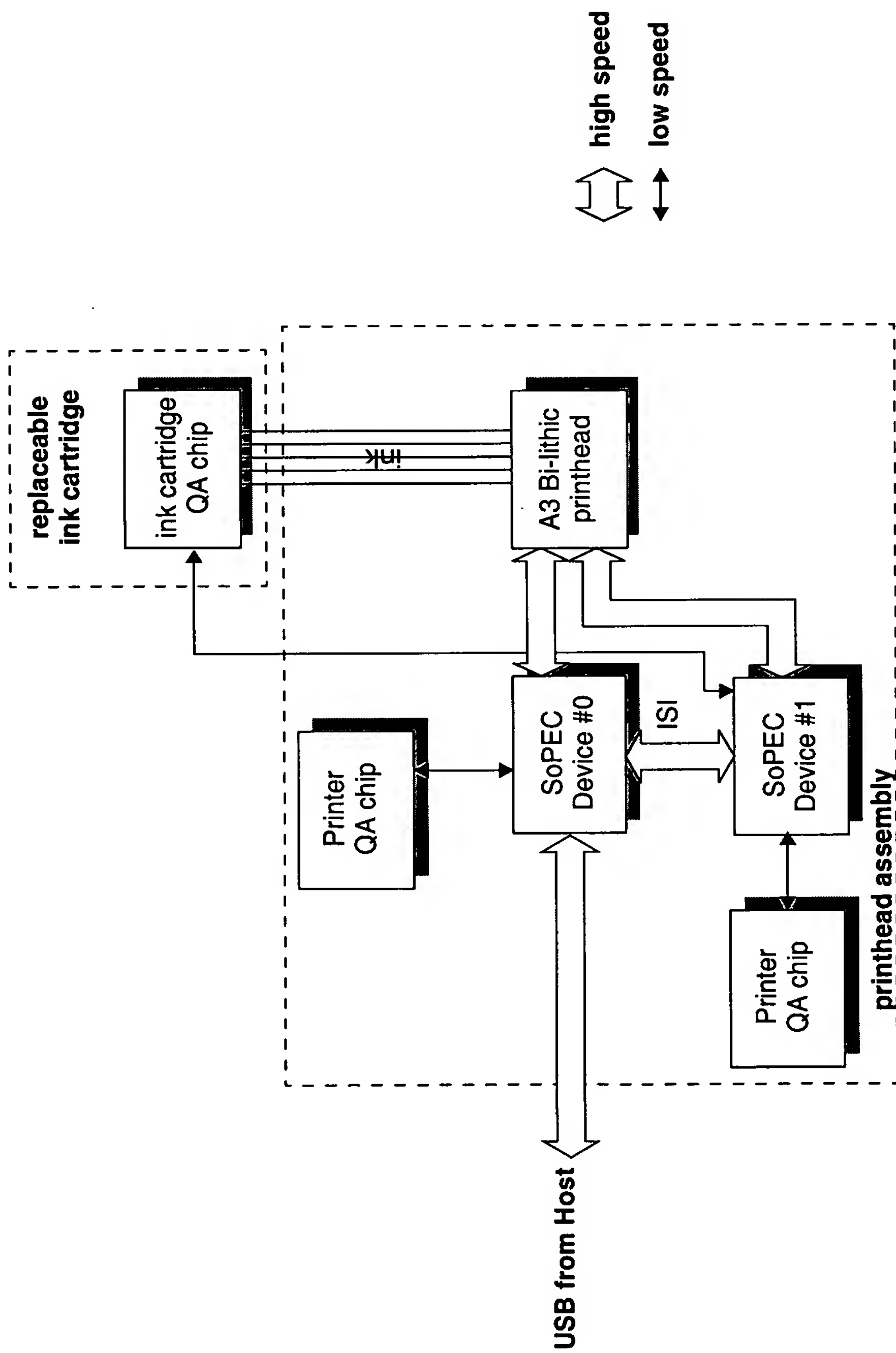


FIG. 5

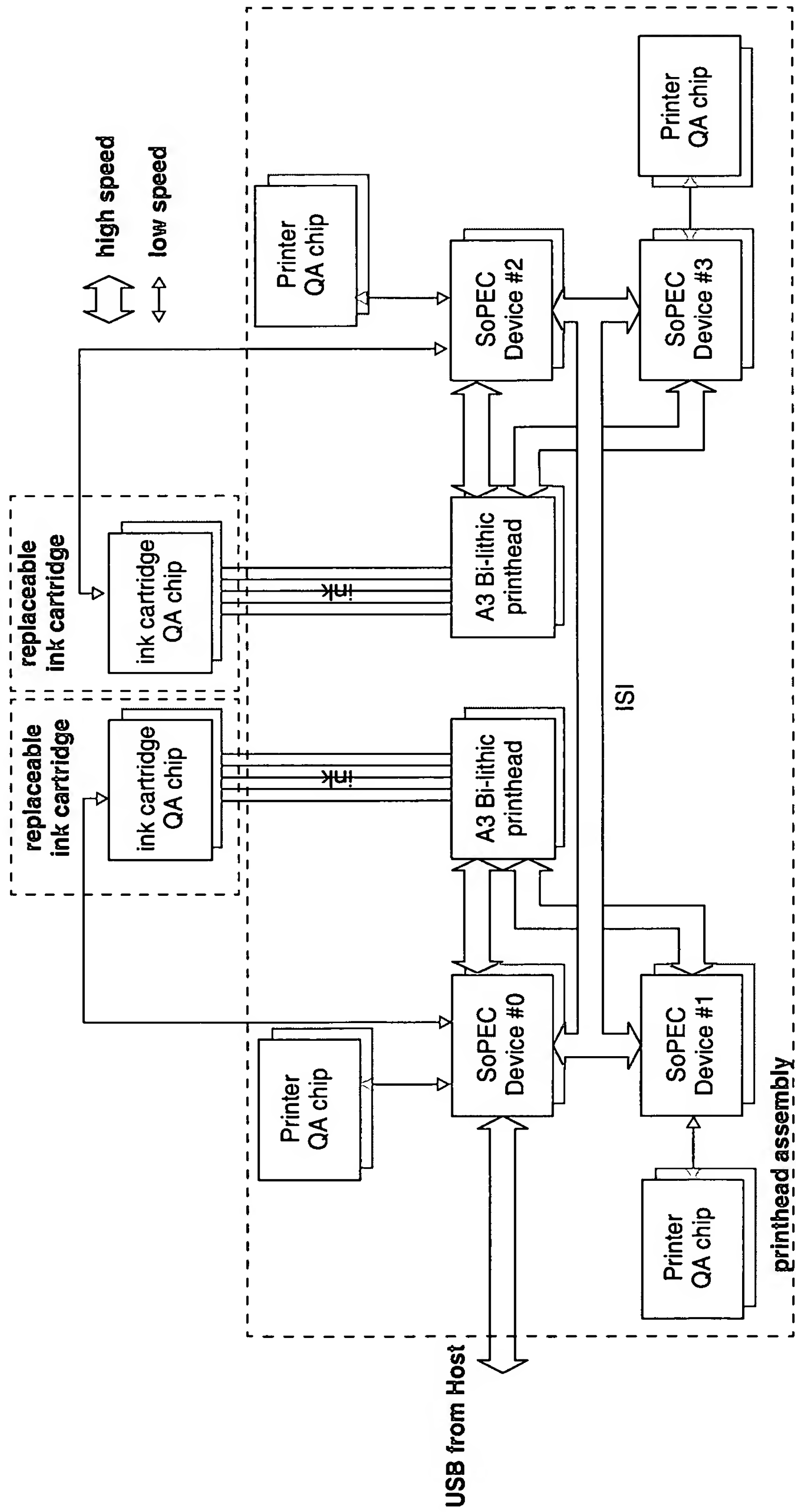


FIG. 6

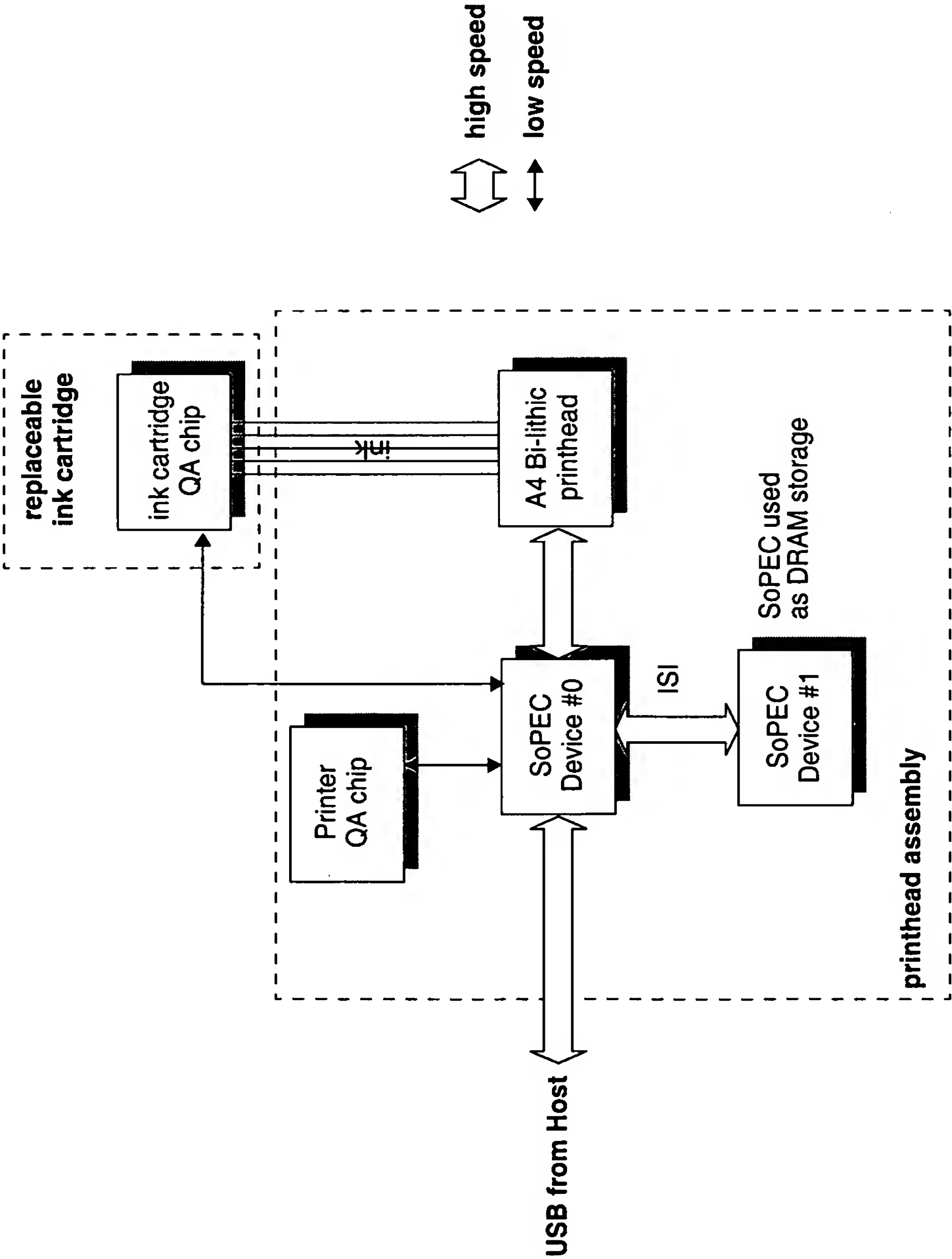


FIG. 7

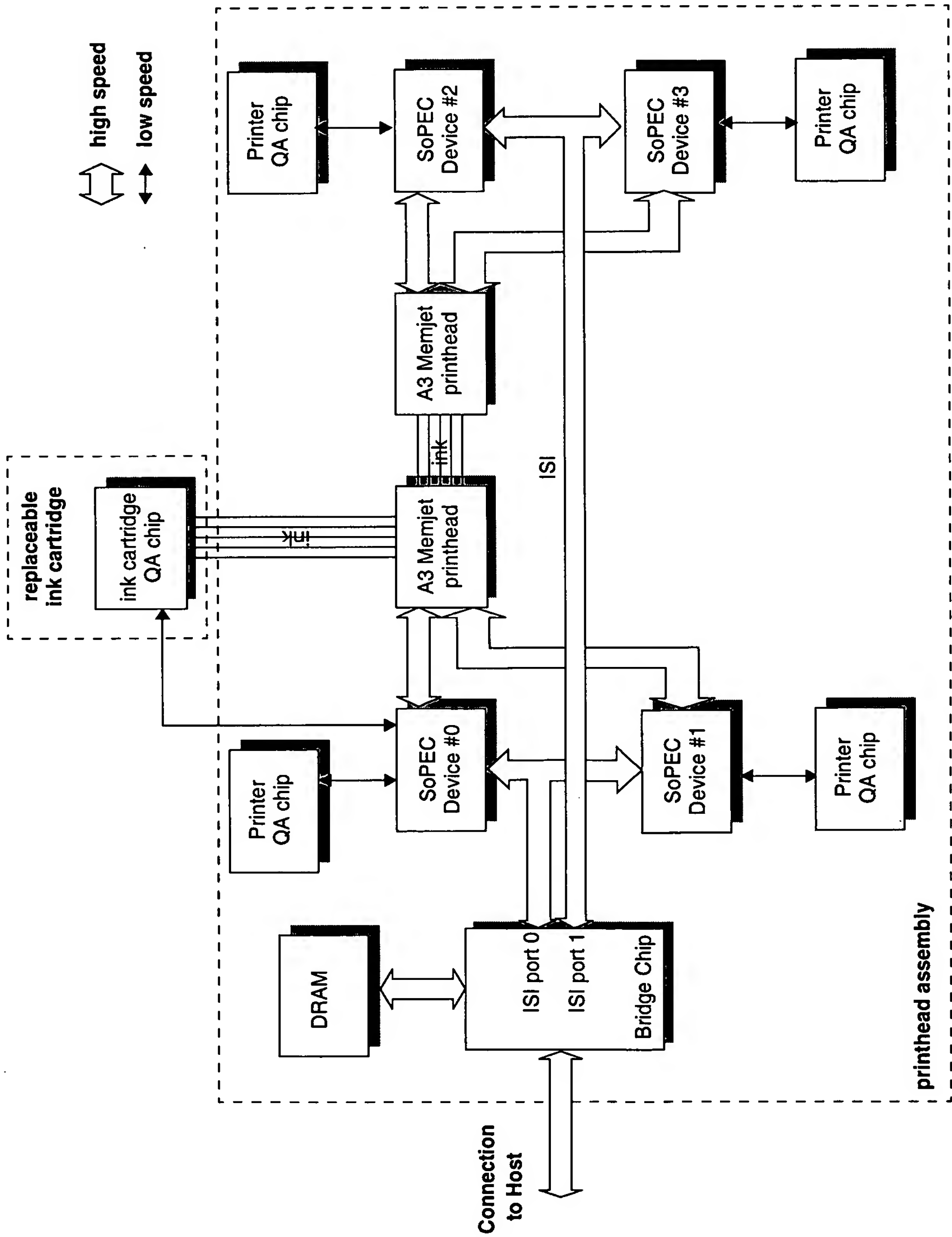


FIG. 8

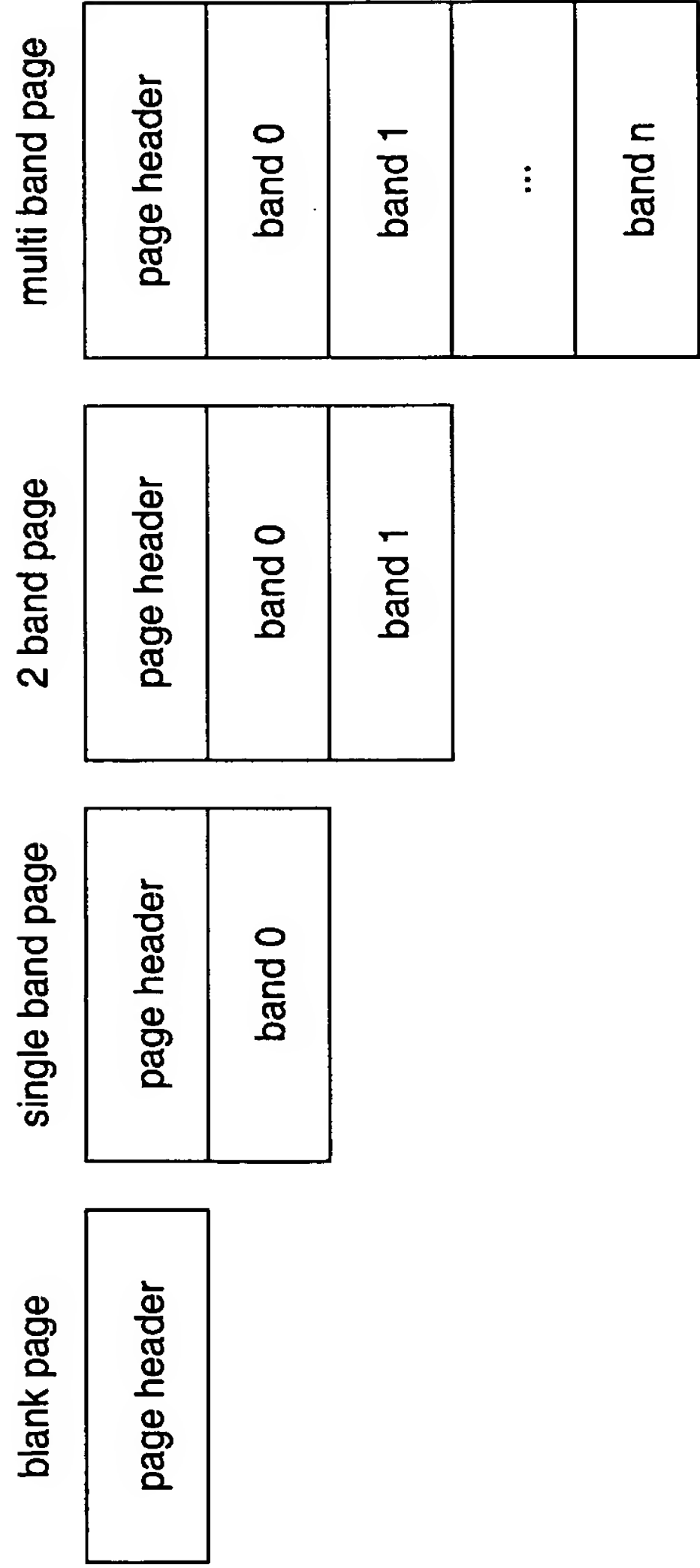


FIG. 9

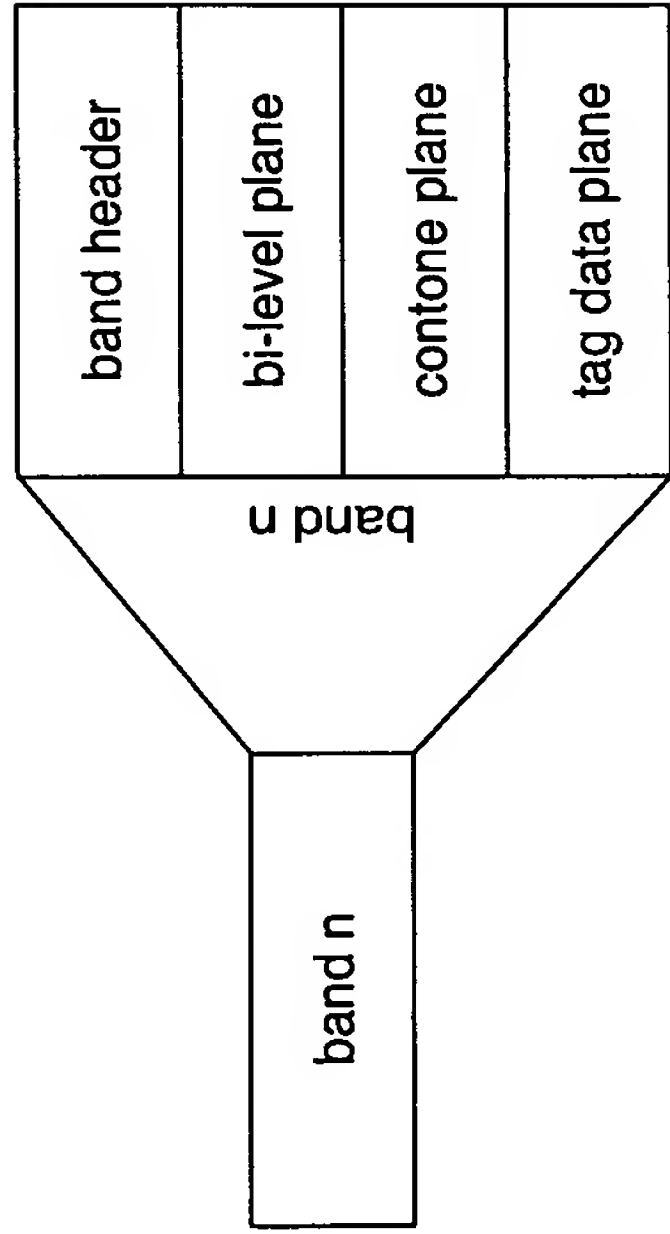


FIG. 10

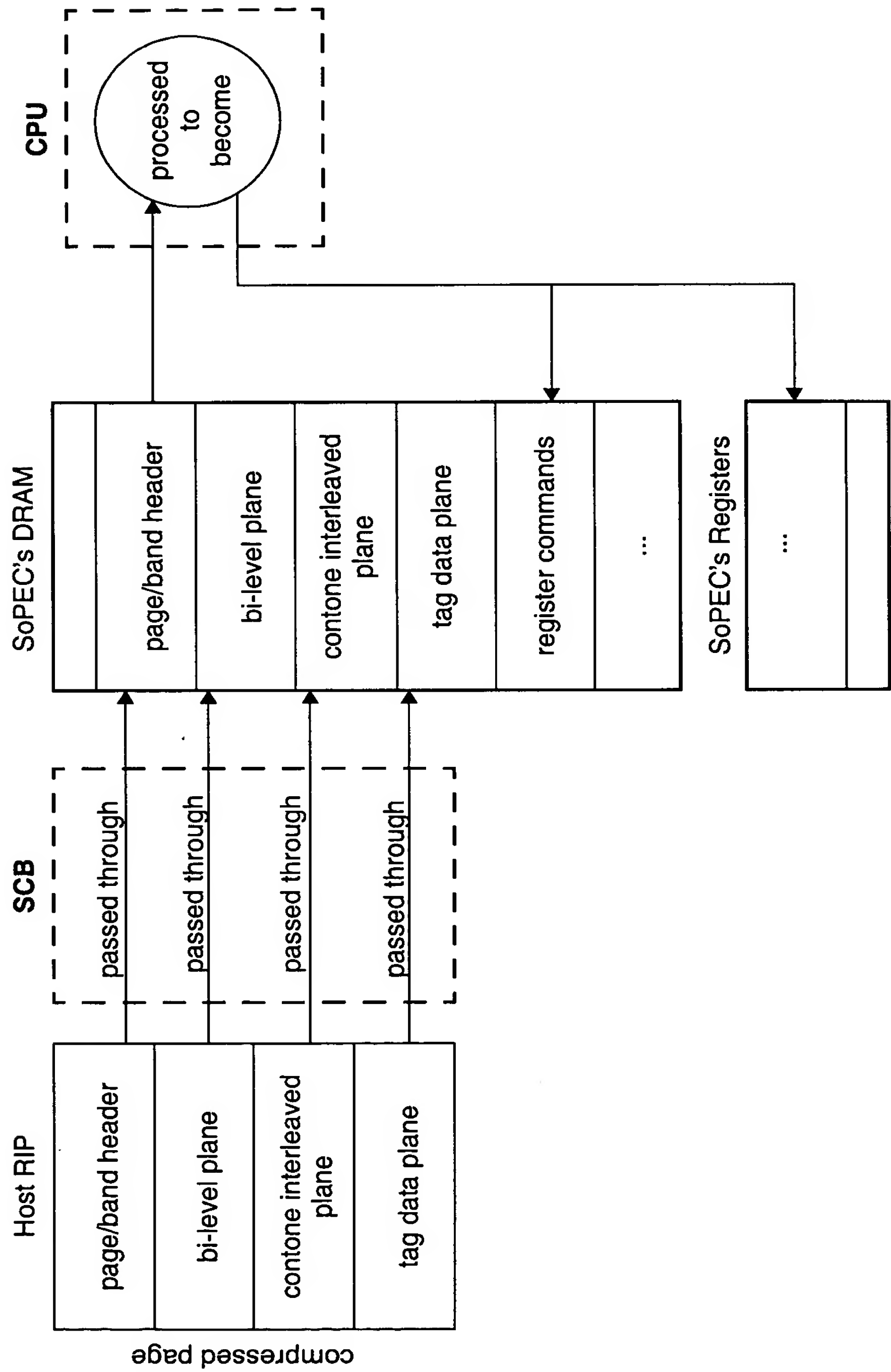


FIG. 11

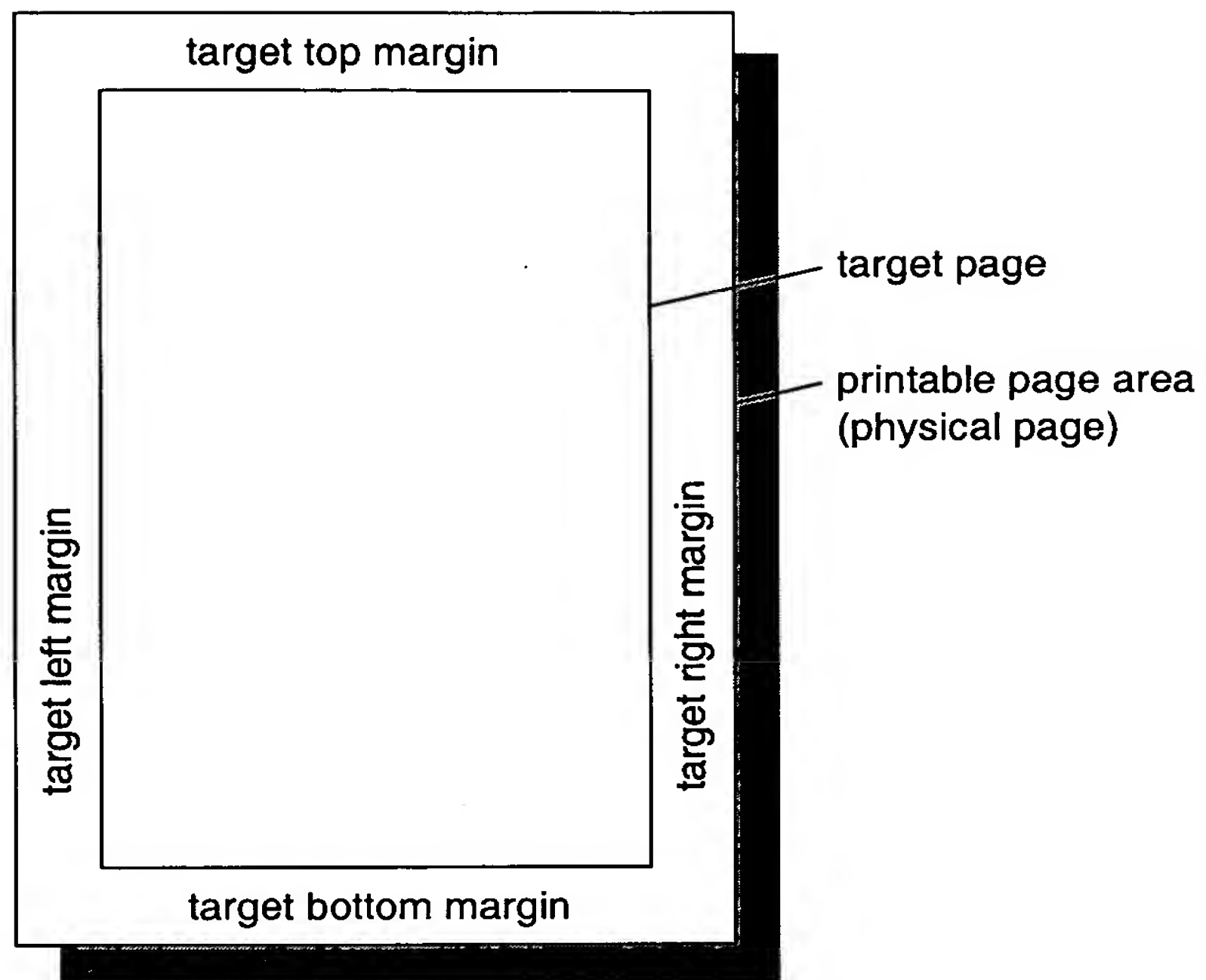


FIG. 12

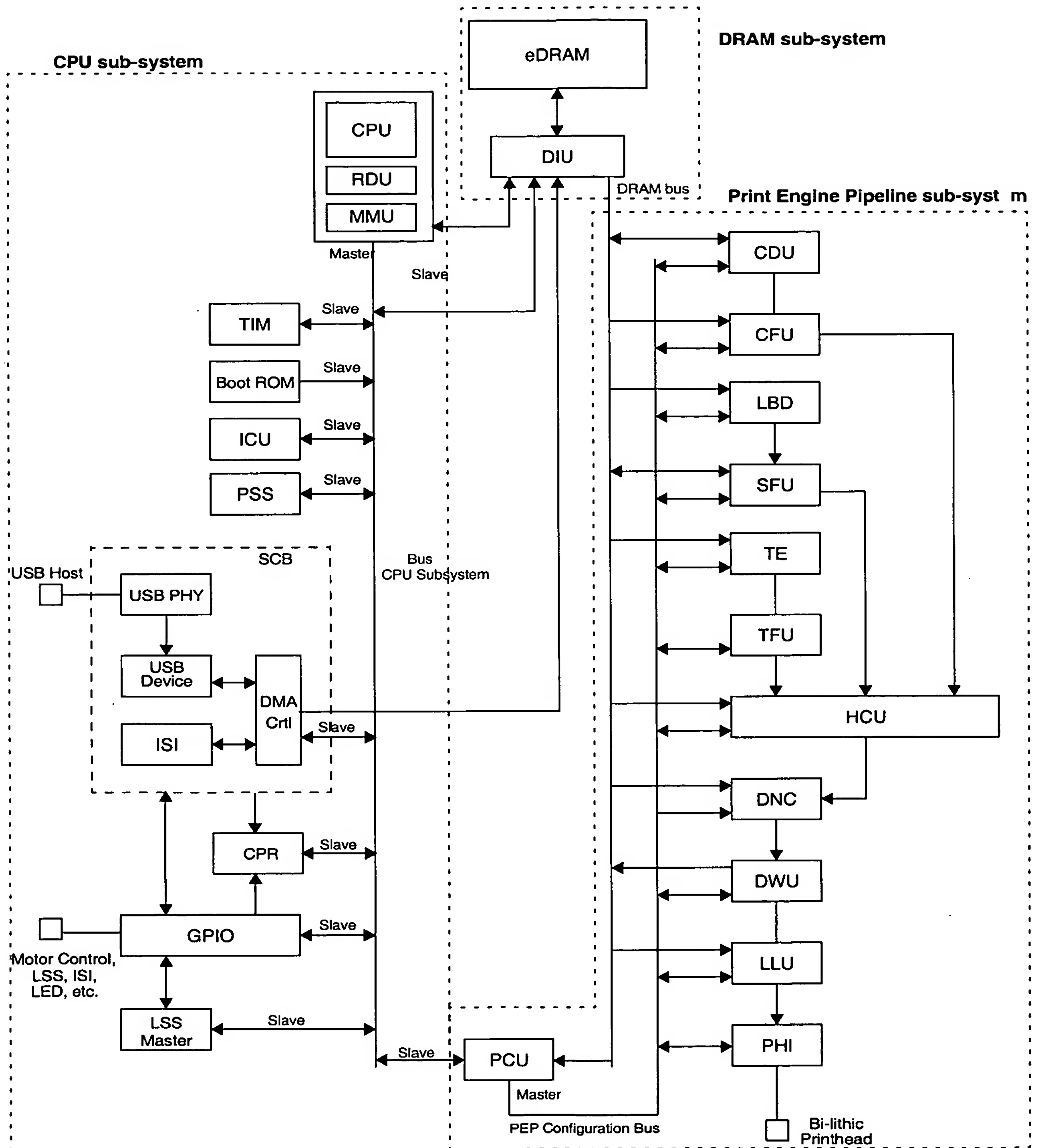


FIG. 13

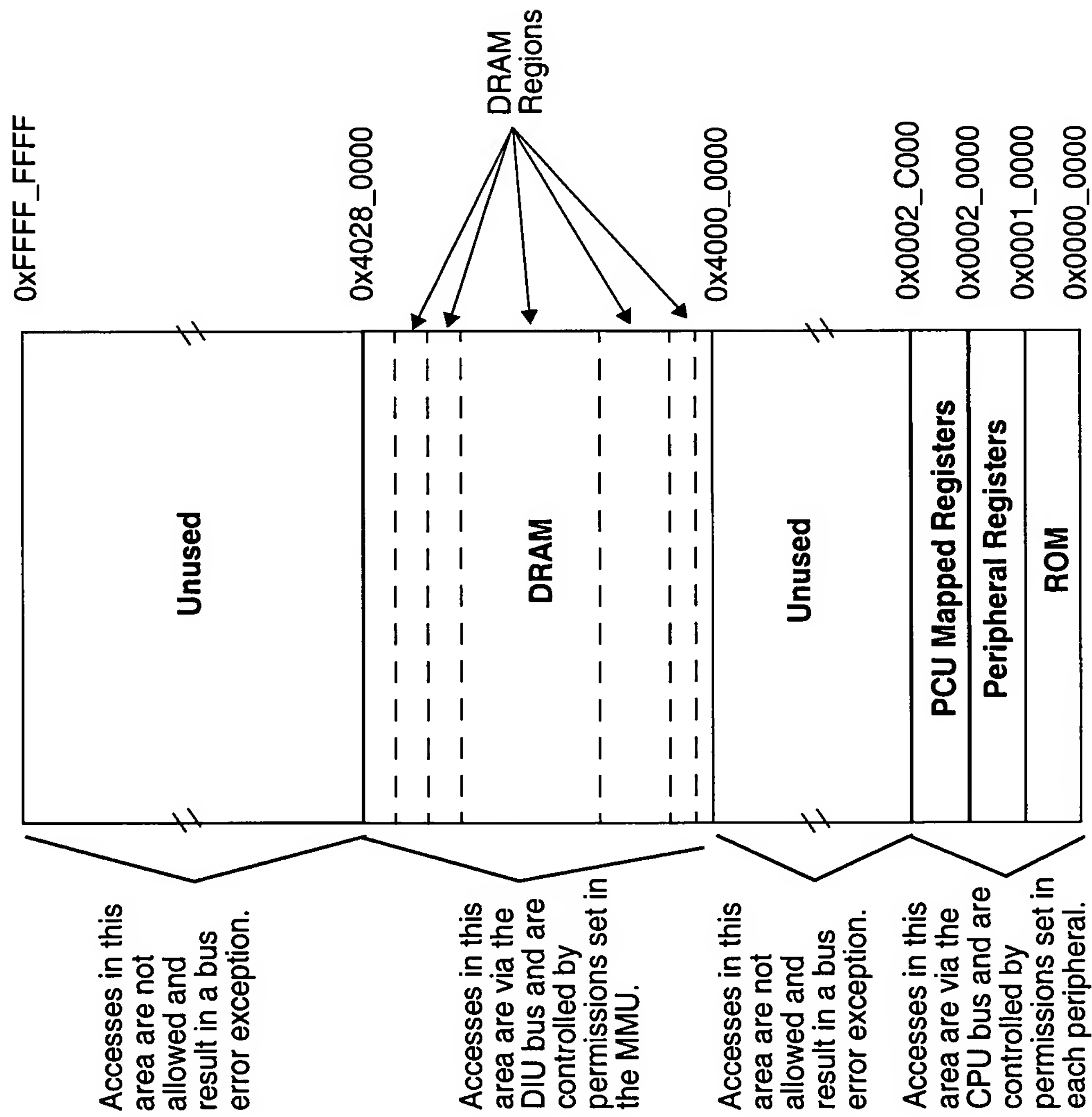


FIG. 14

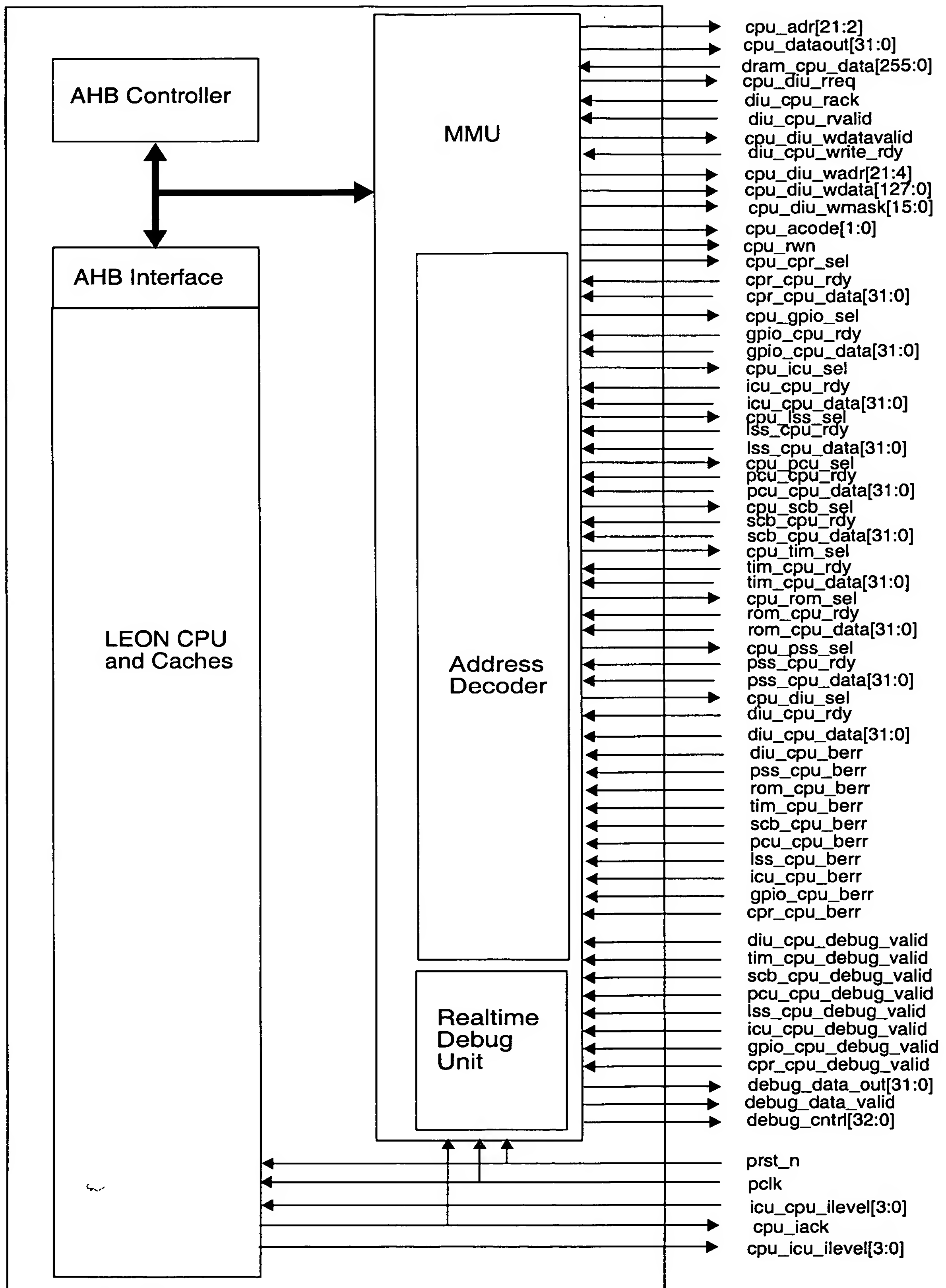


FIG. 15

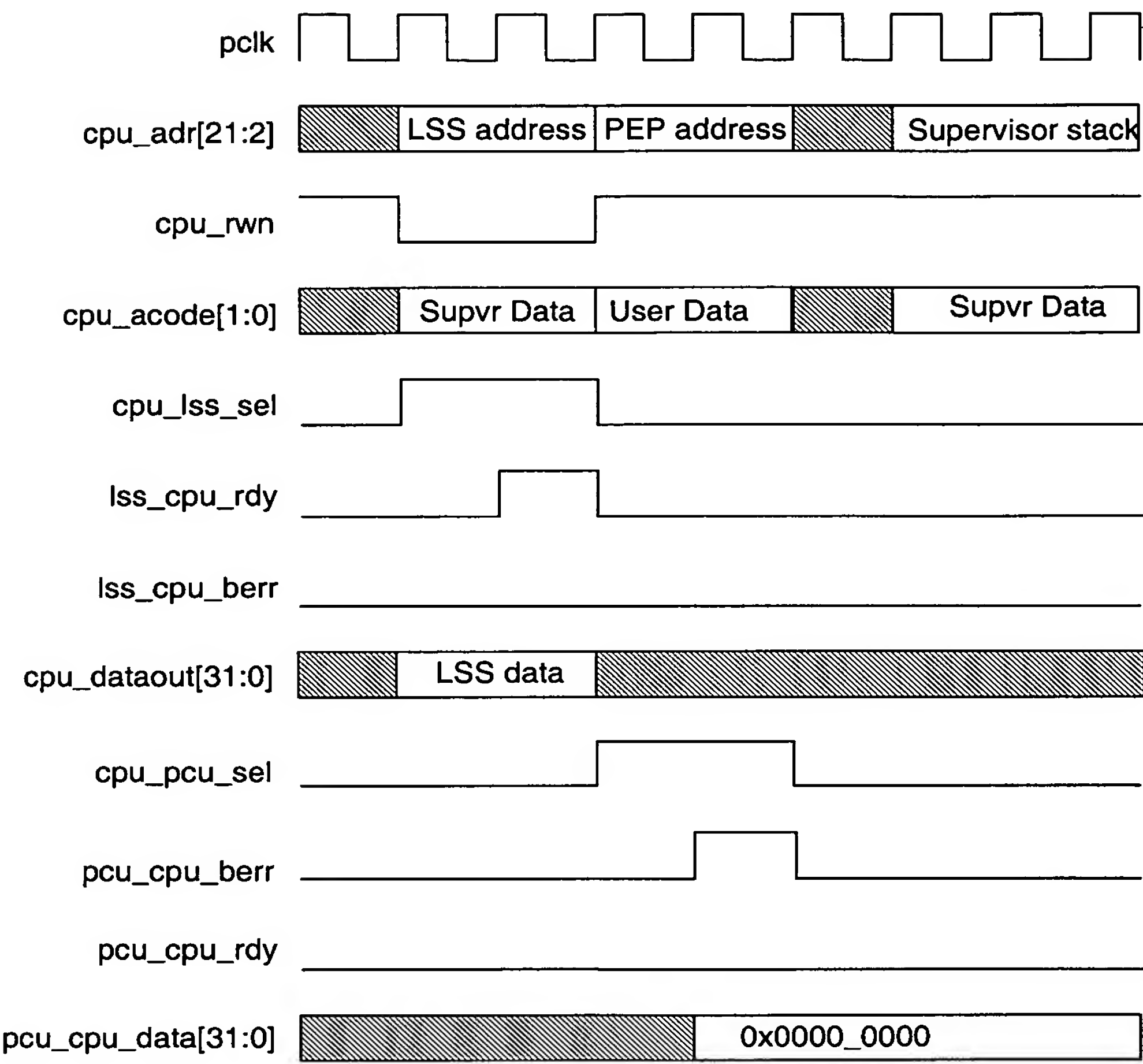


FIG. 16

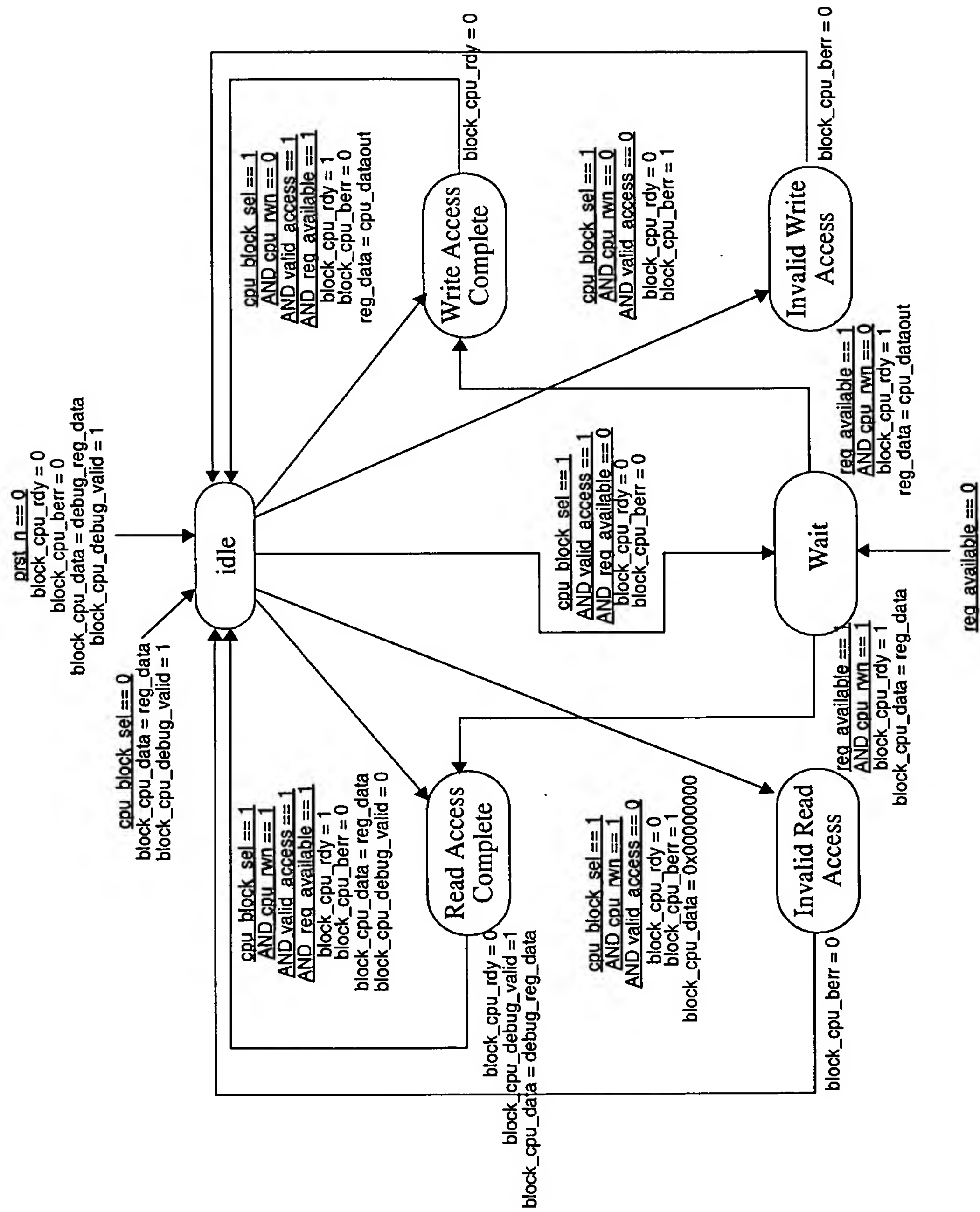


FIG. 17

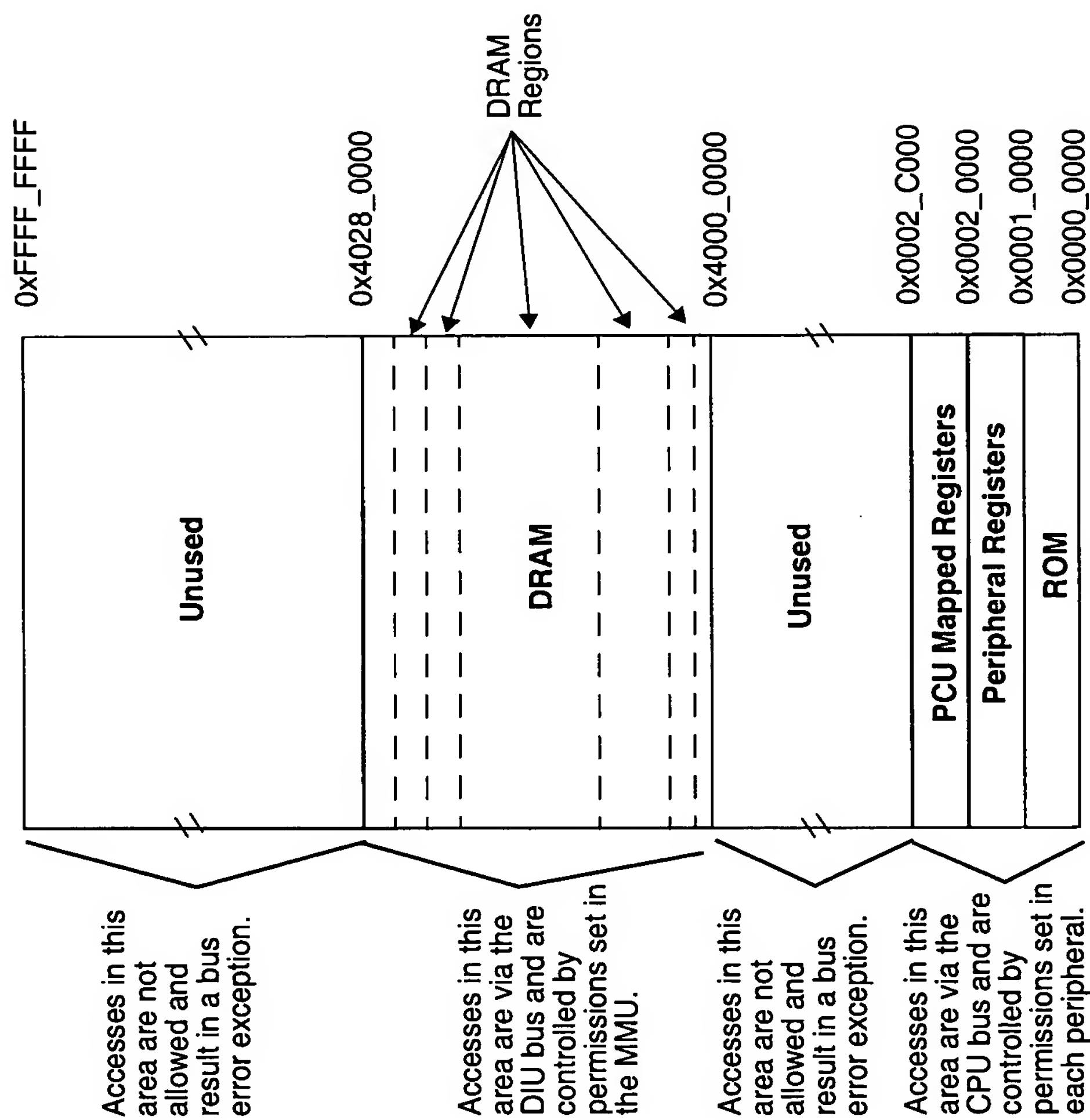


FIG. 18

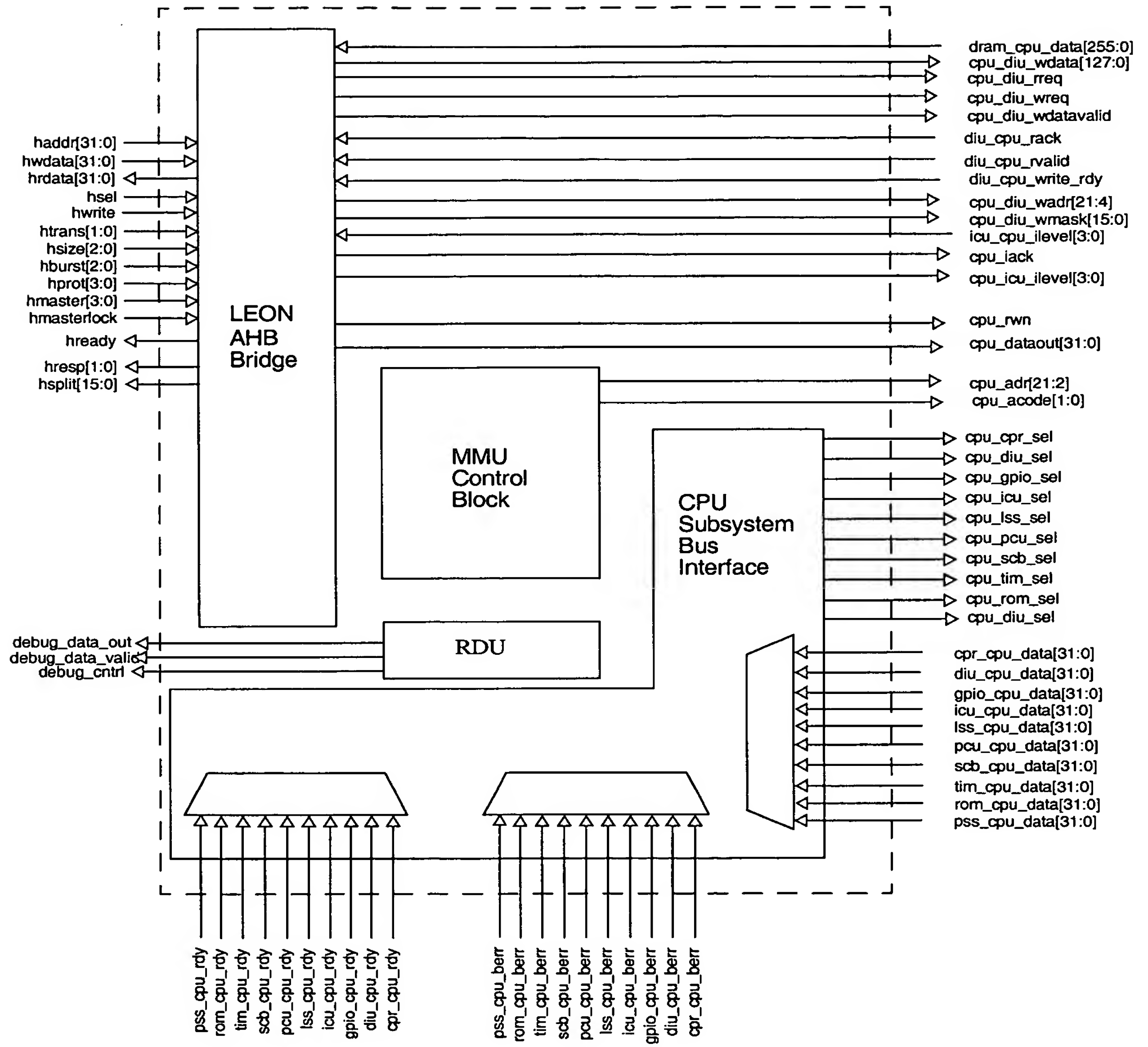


FIG. 19

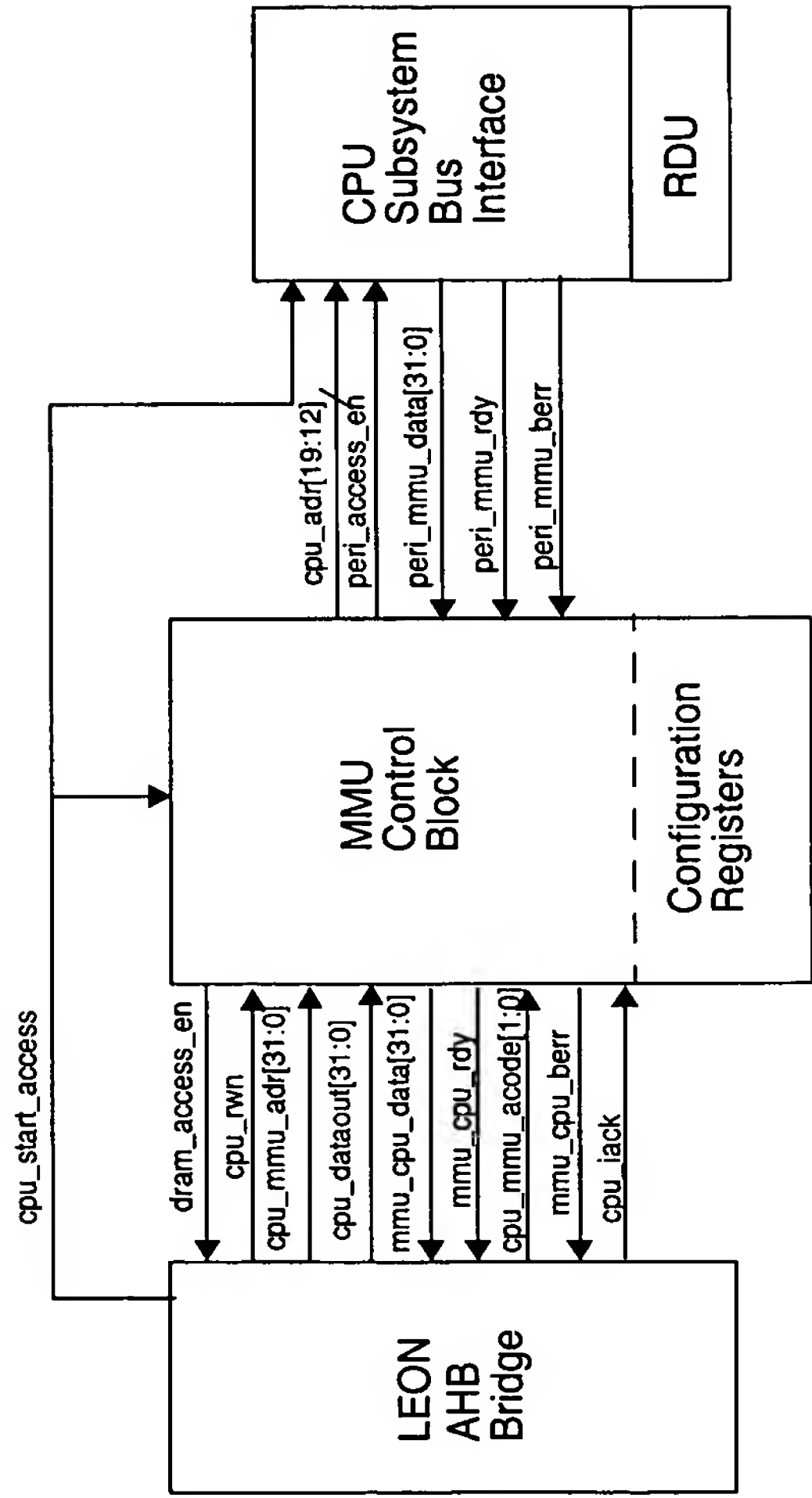


FIG. 20

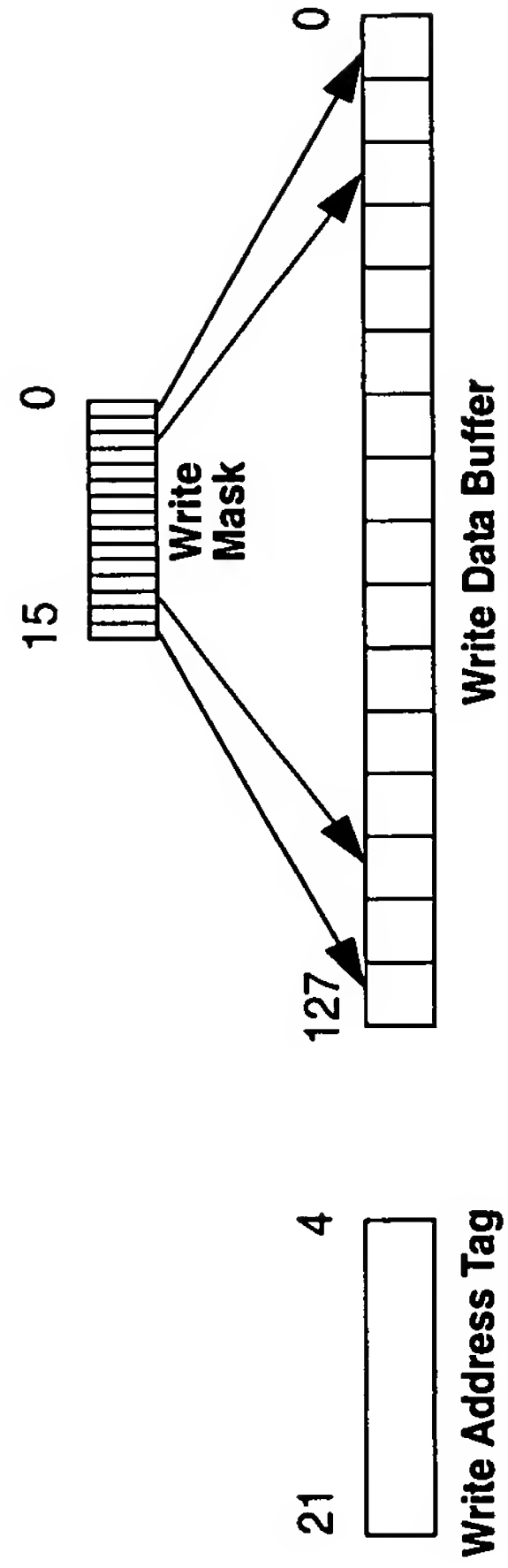


FIG. 21

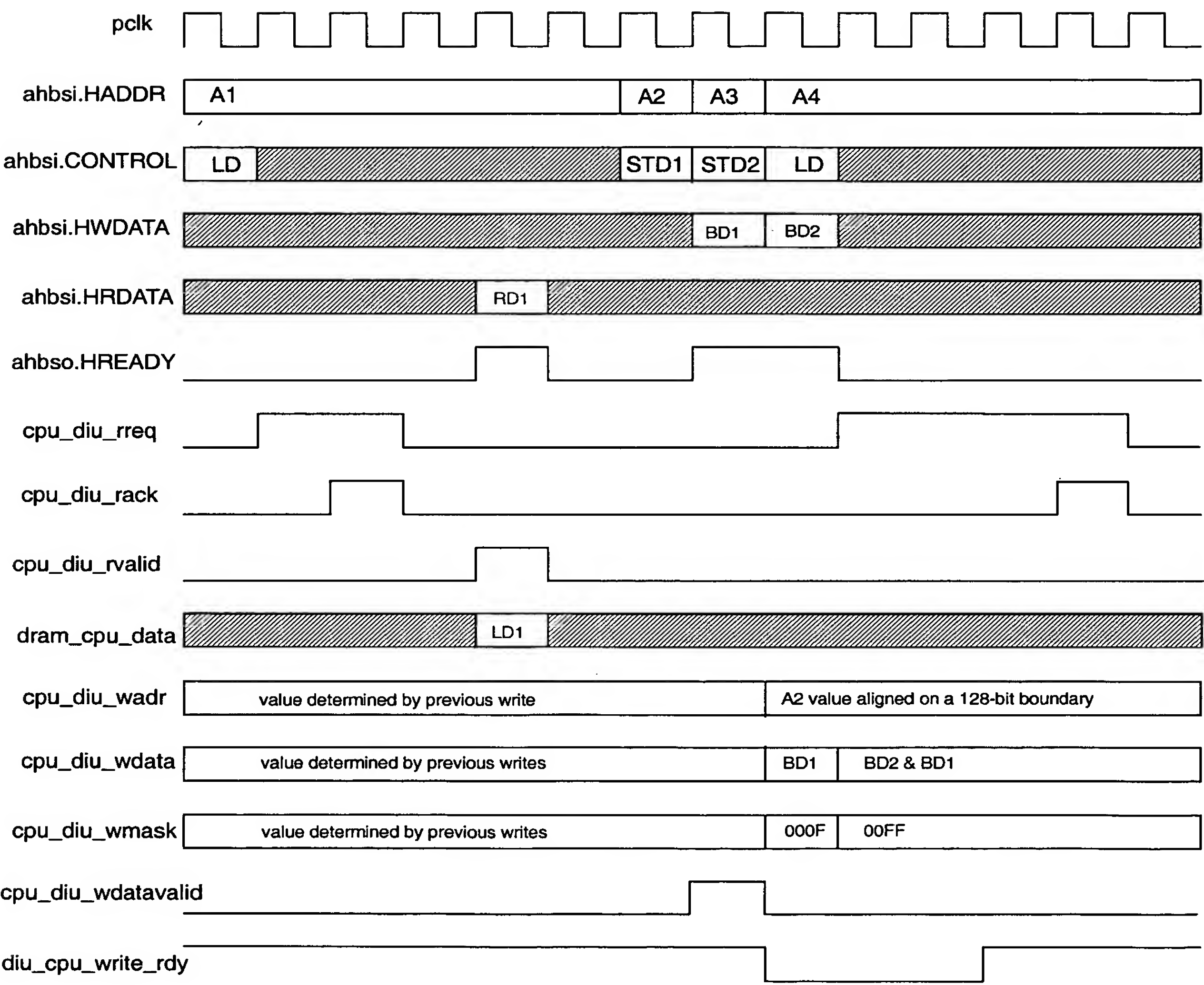


FIG. 22

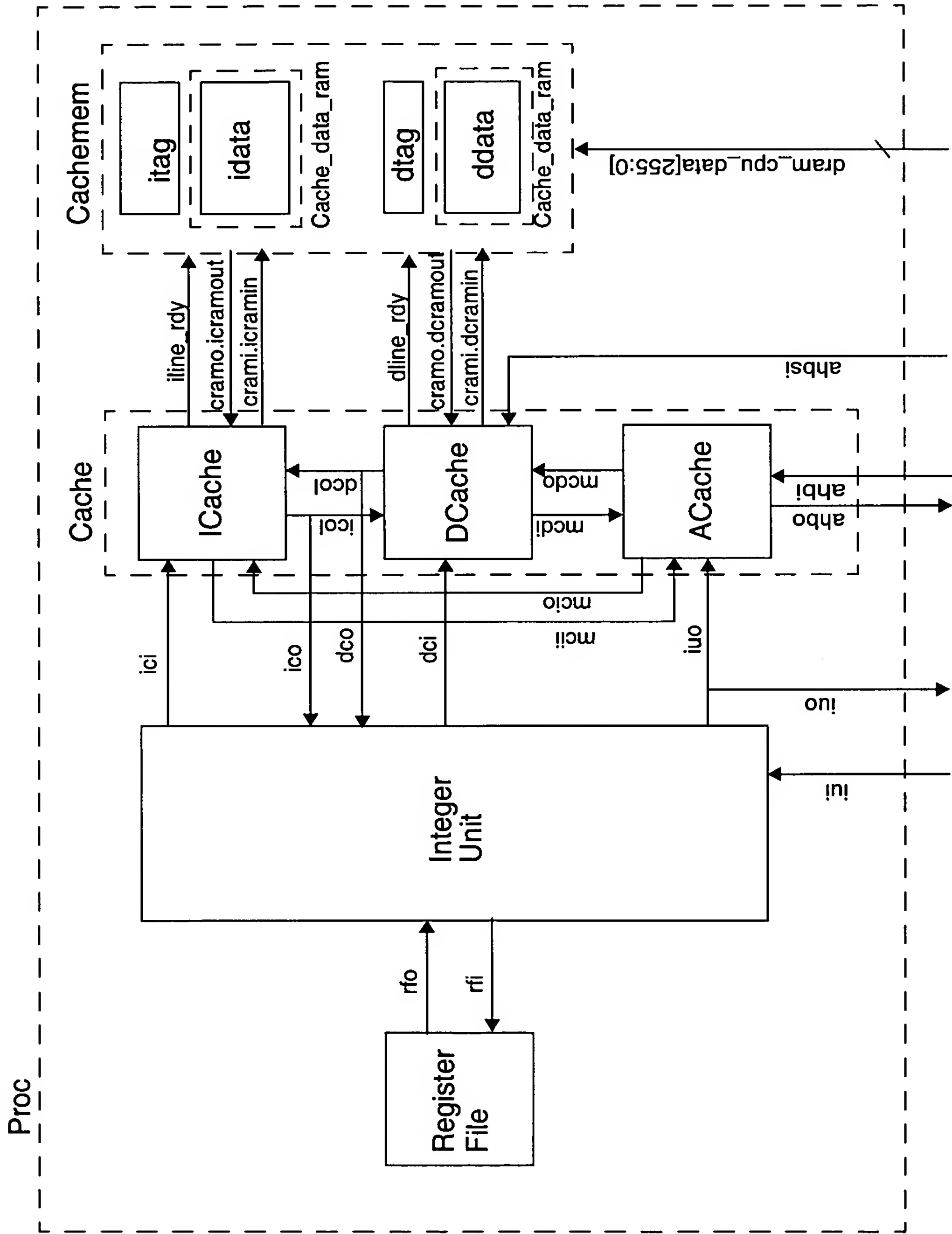


FIG. 23

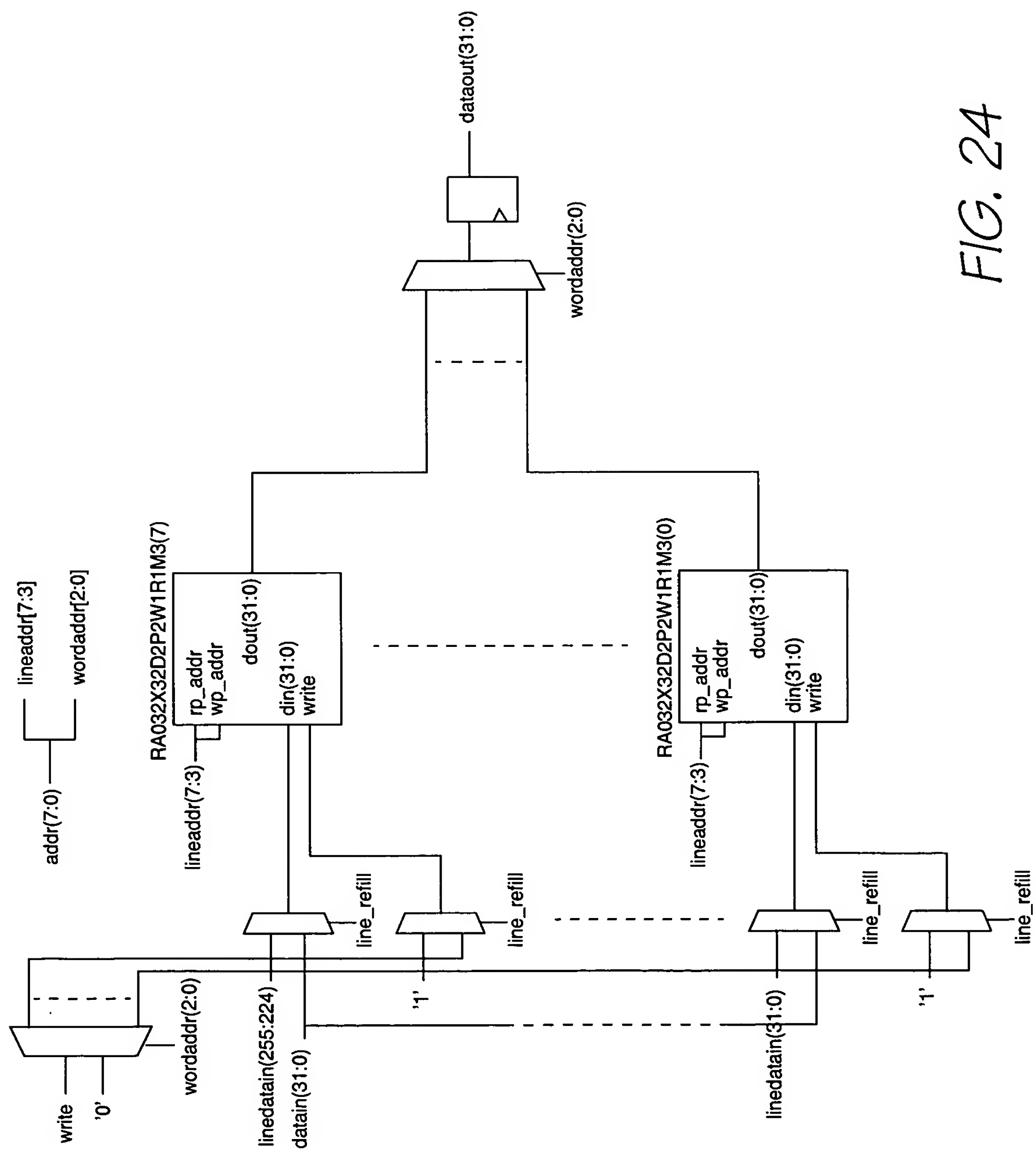


FIG. 24

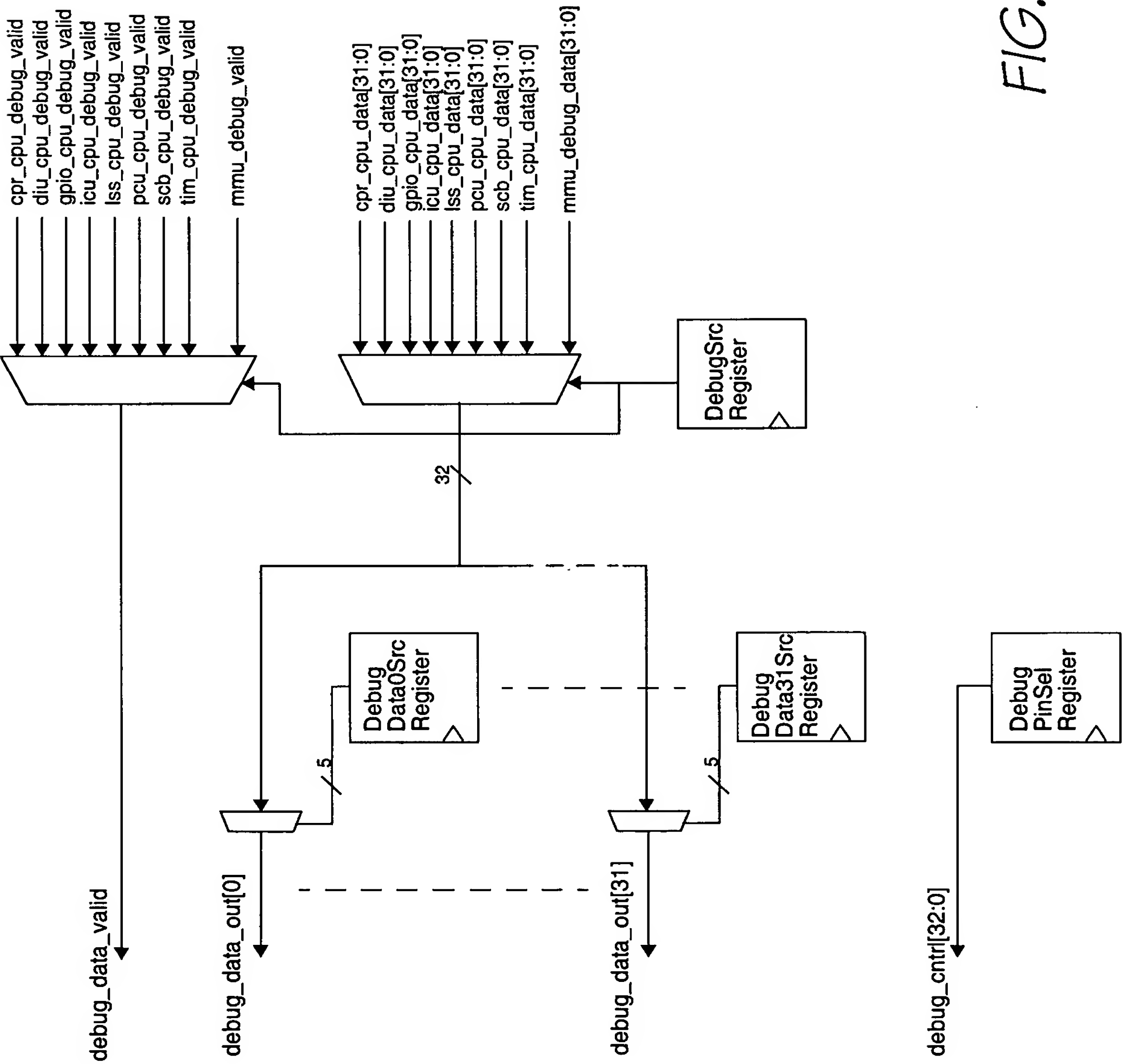


FIG. 25

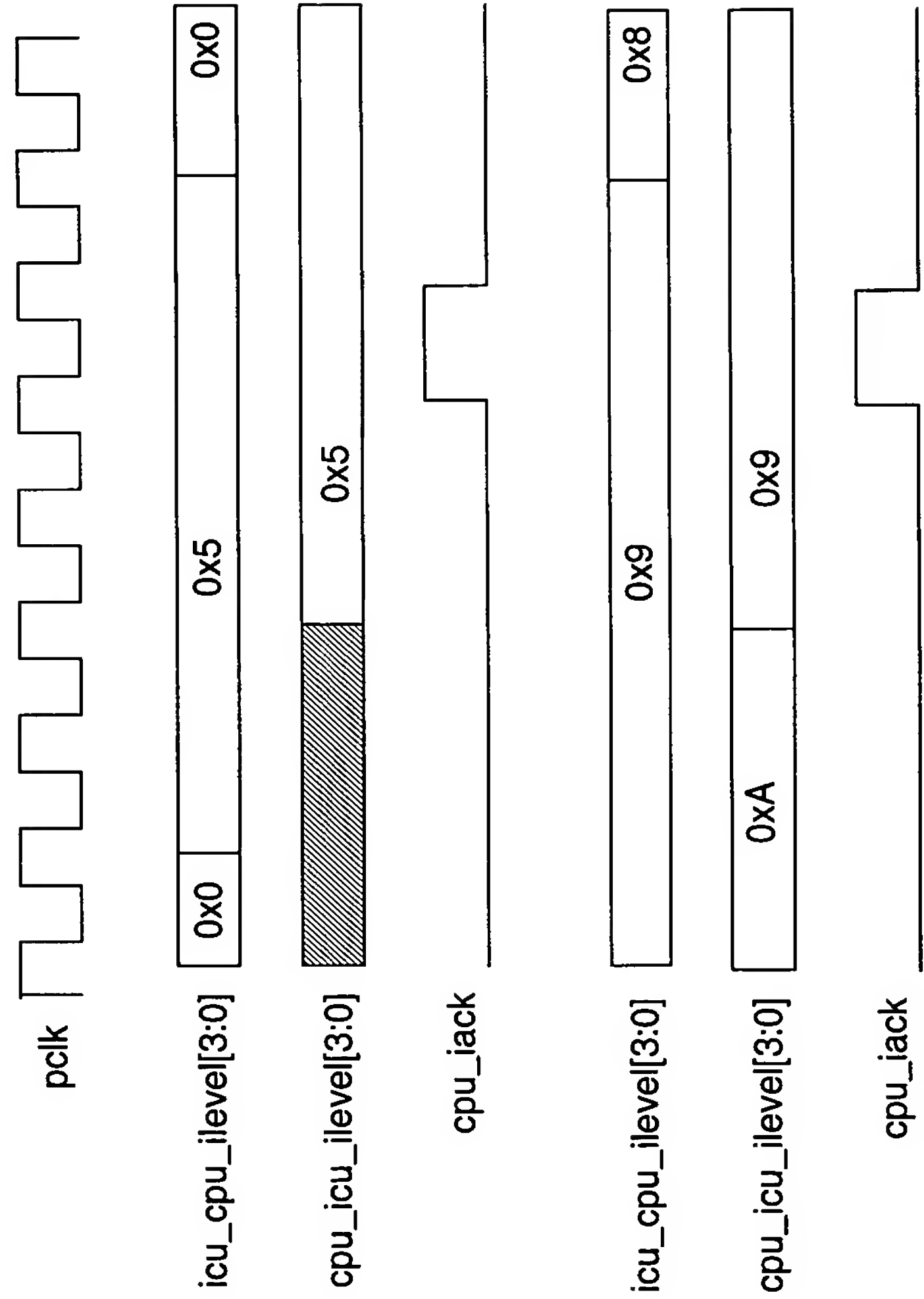


FIG. 26

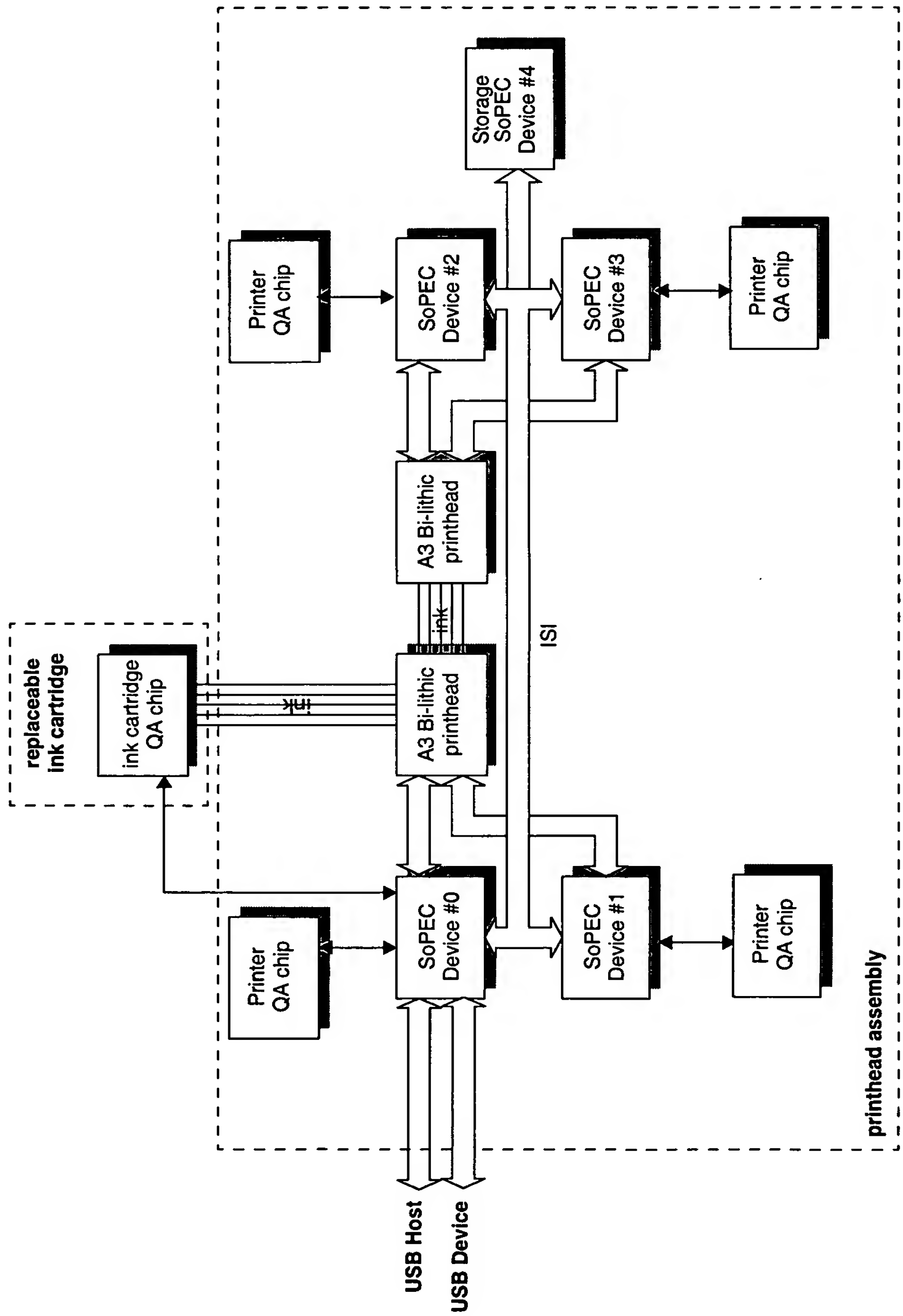


FIG. 27

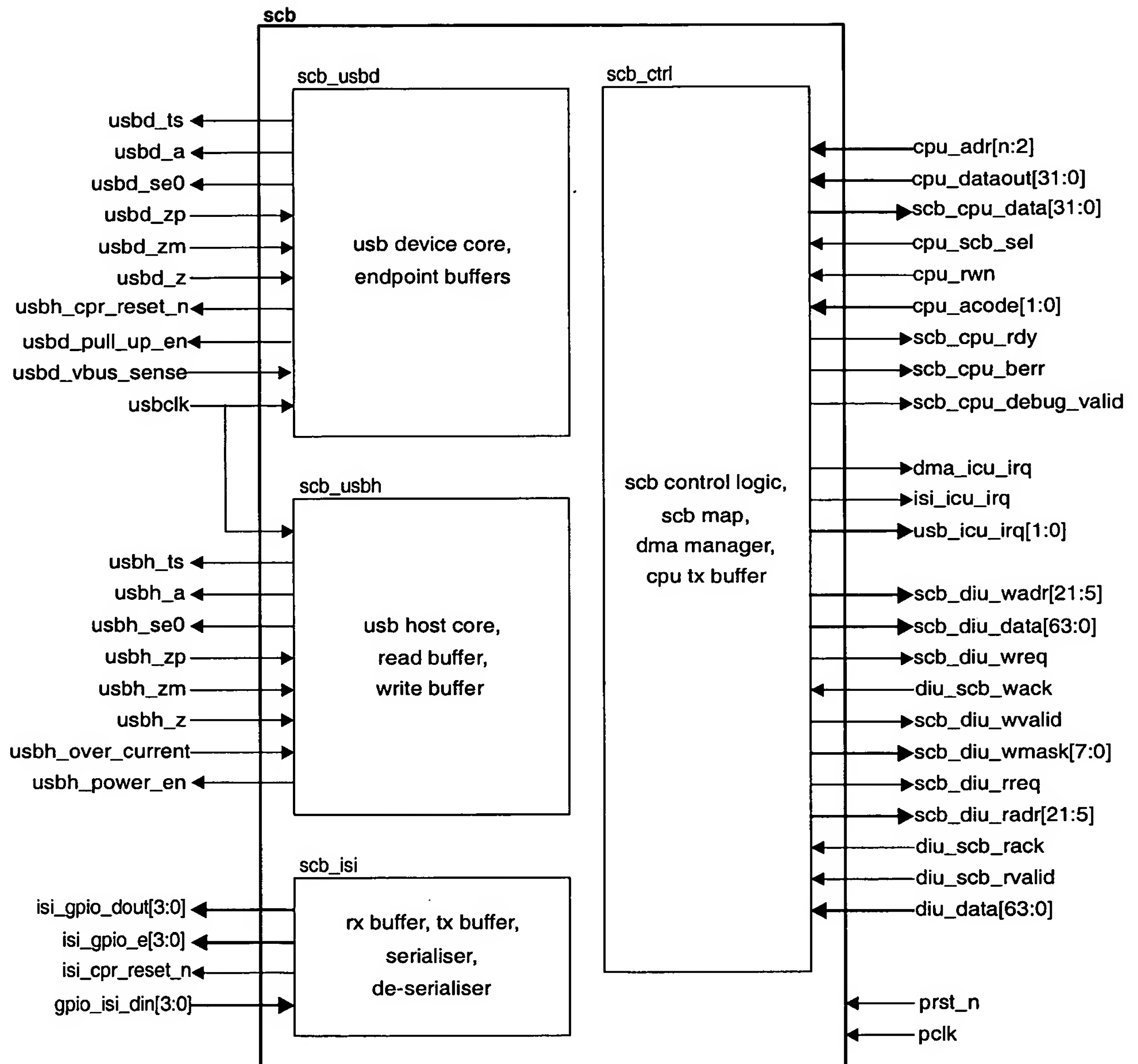


FIG. 28

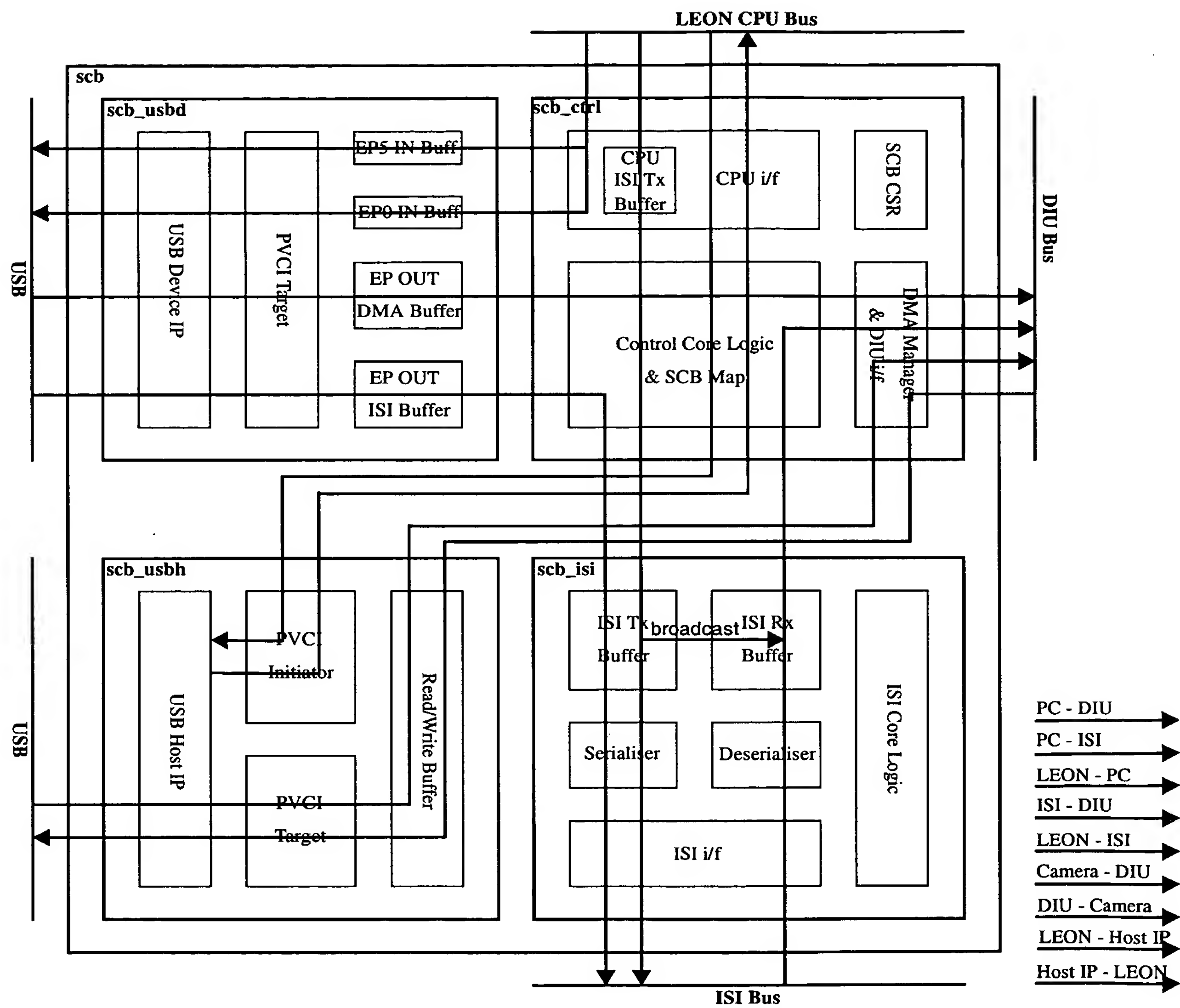


FIG. 29

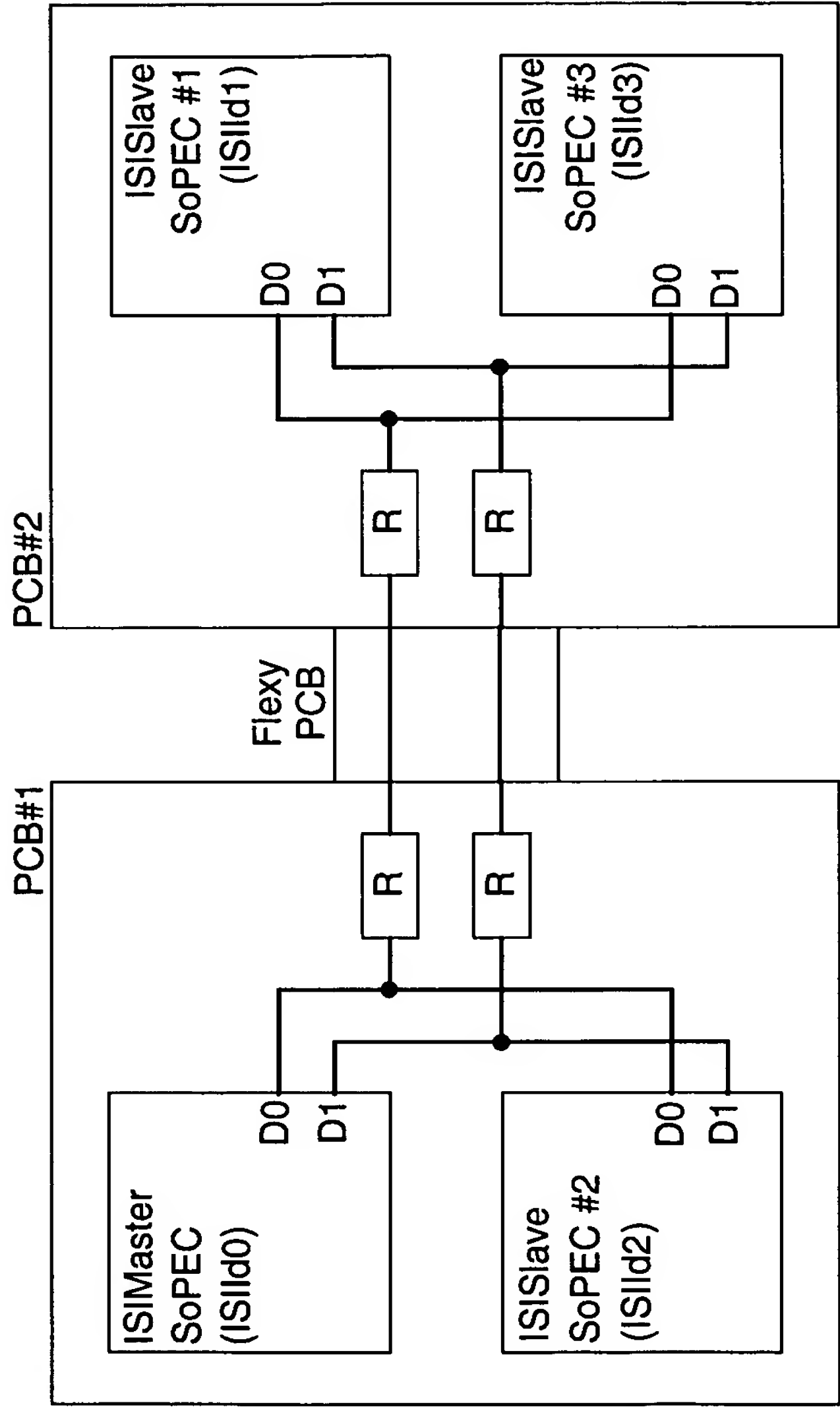


FIG. 30

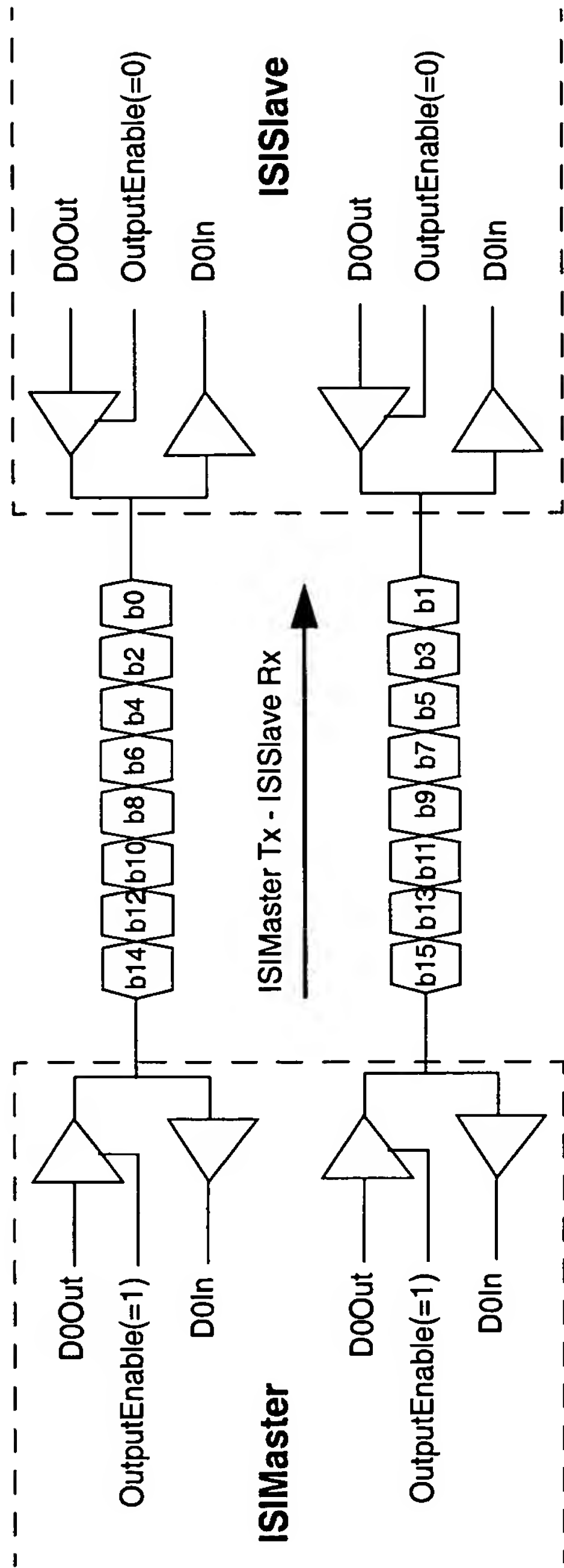
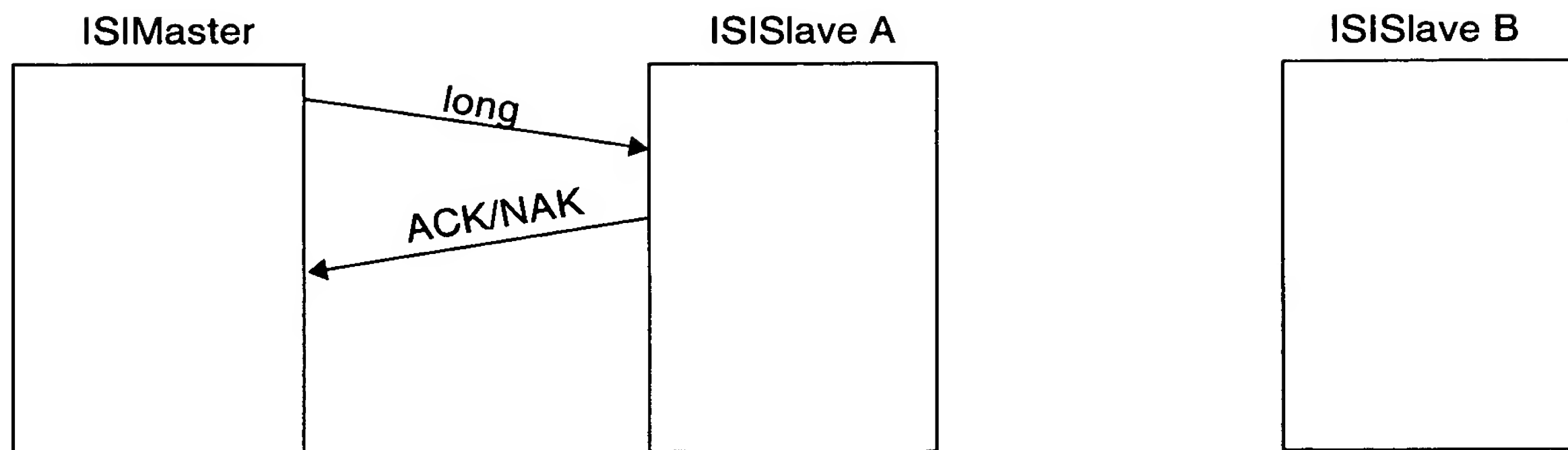
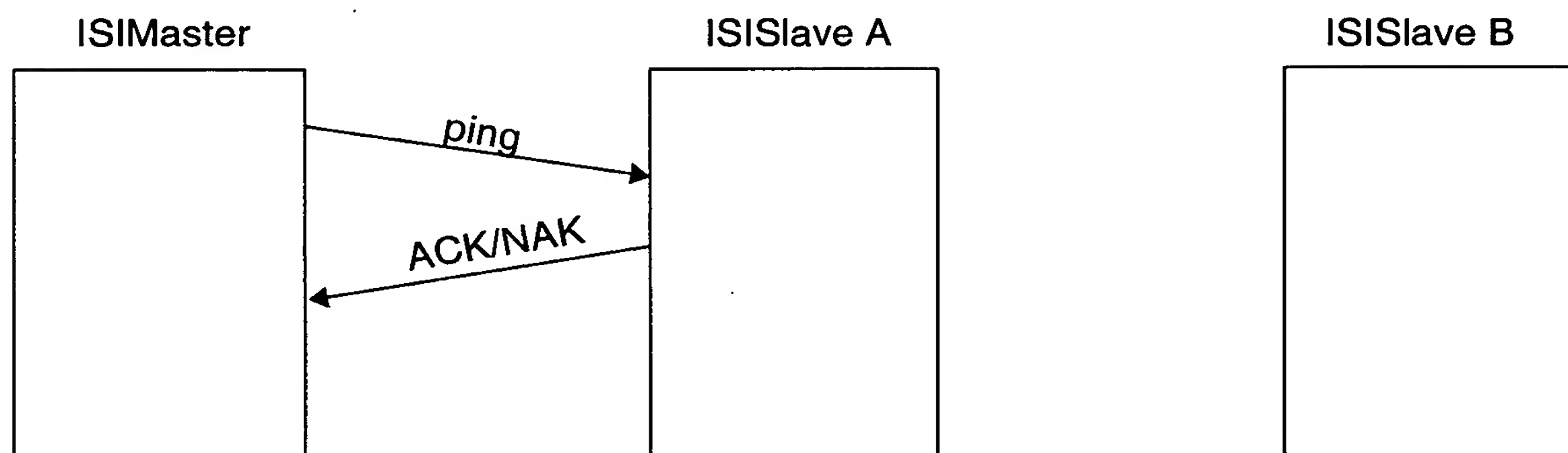


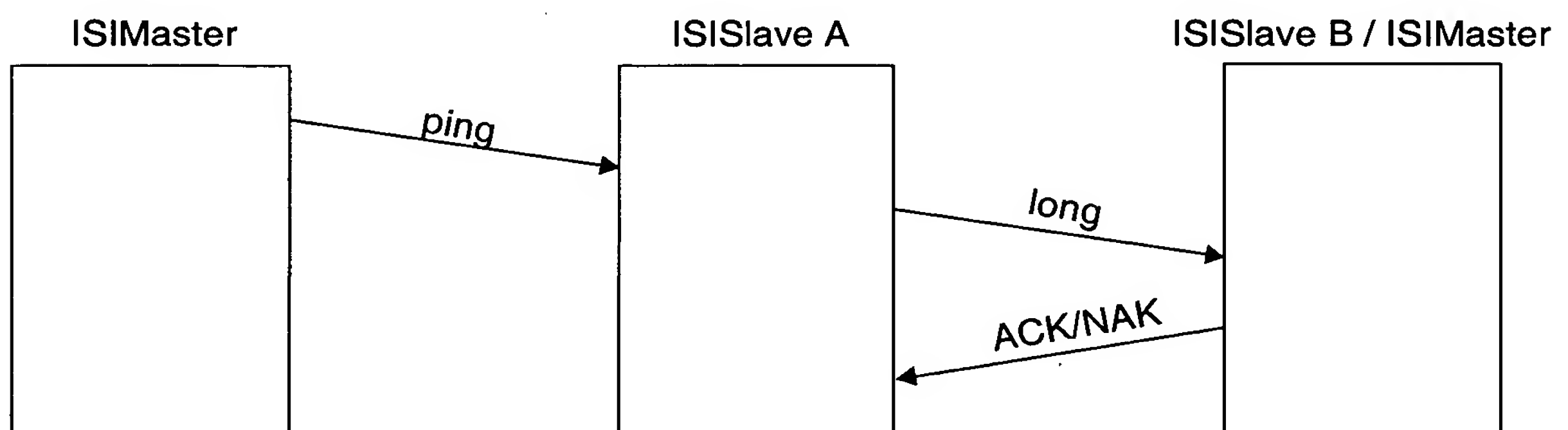
FIG. 31



Transaction 1: Long packet to an addressed ISISlave



Transaction 2: Ping packet to an addressed ISISlave. ISISlave has nothing to send



Transaction 3: Ping packet to an addressed ISISlave. ISISlaveA responds with a long packet to ISISlaveB (or the ISIMaster) and ISISlaveB (or the ISIMaster) responds with an ACK or NAK.

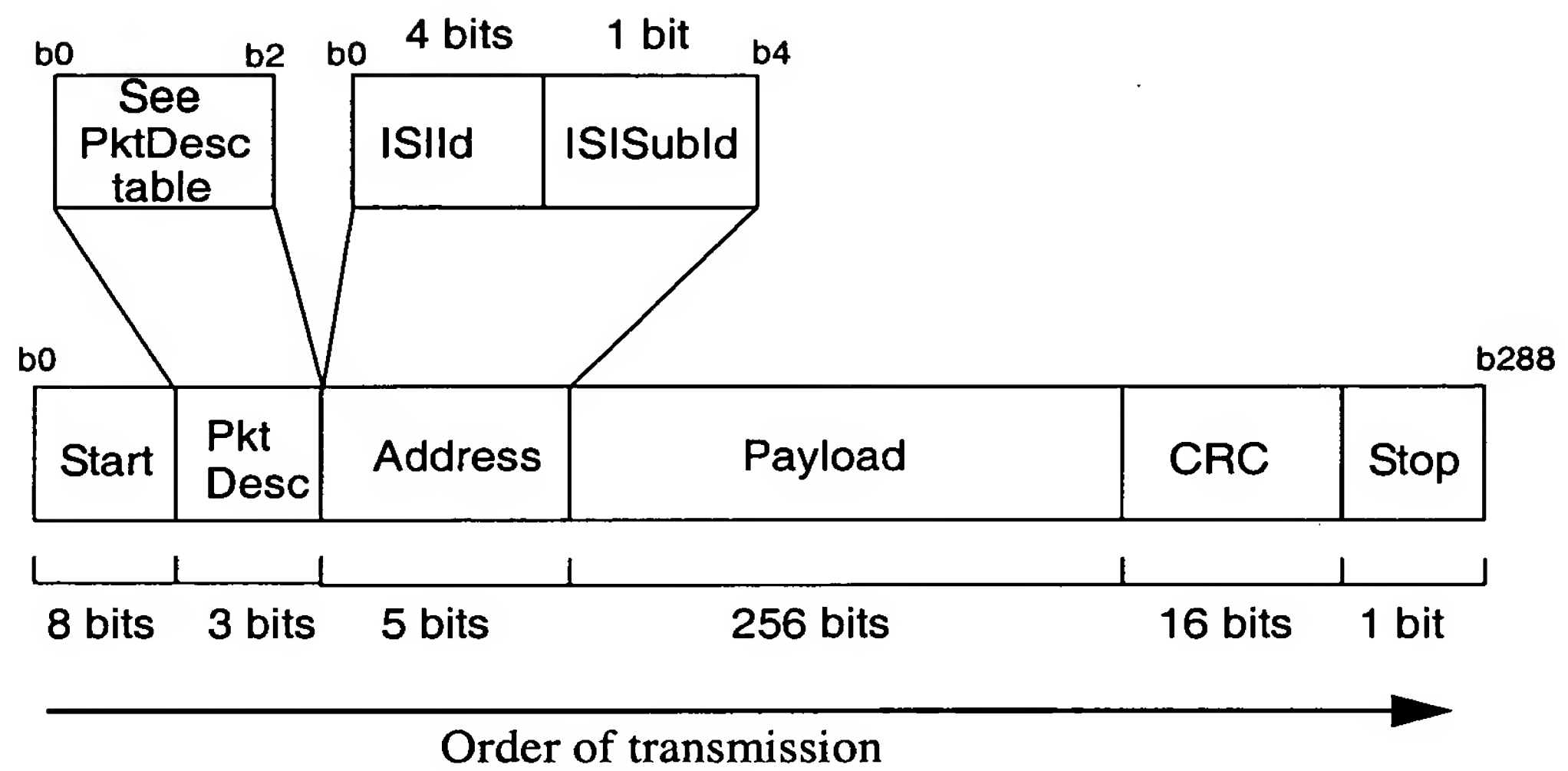


FIG. 33

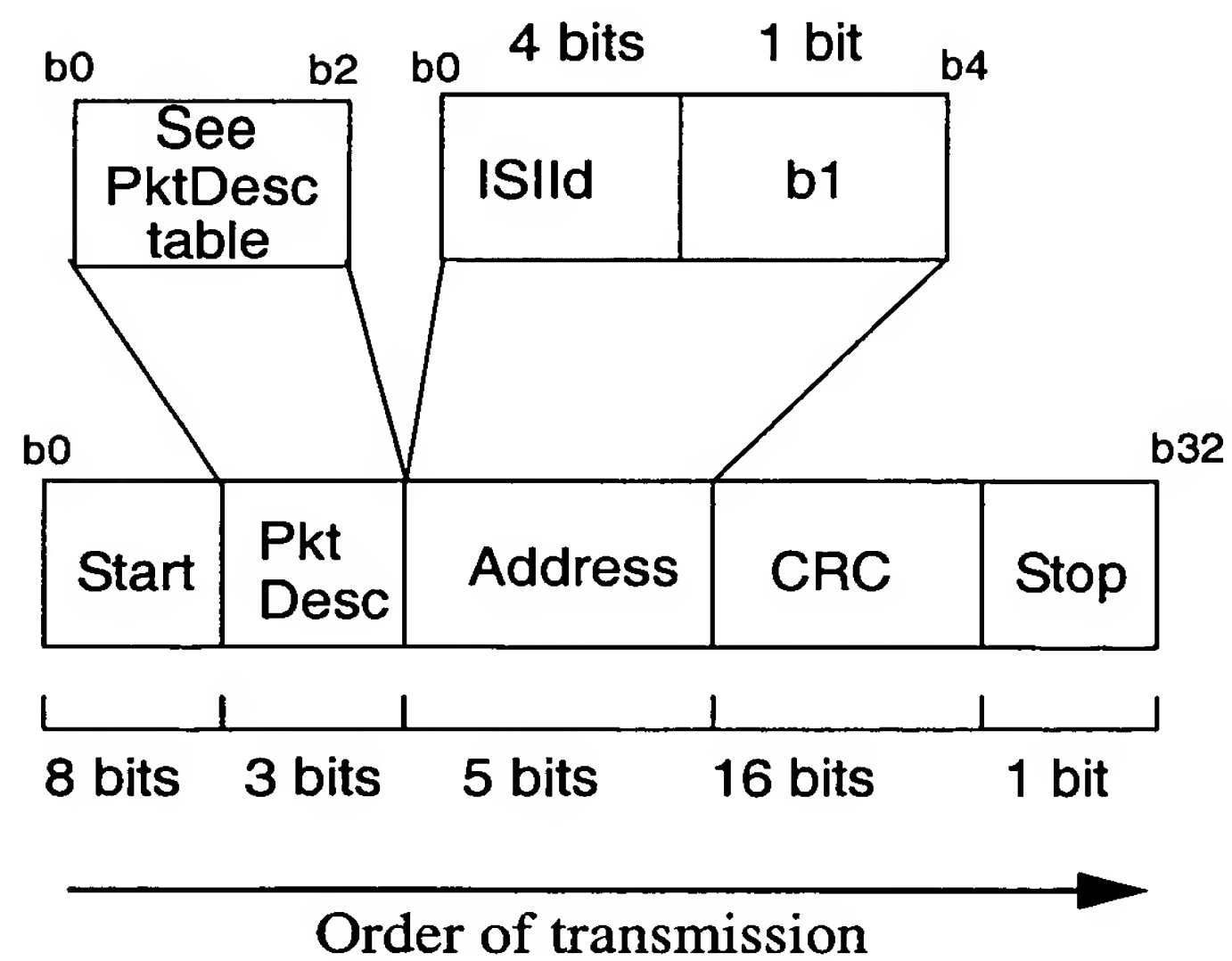
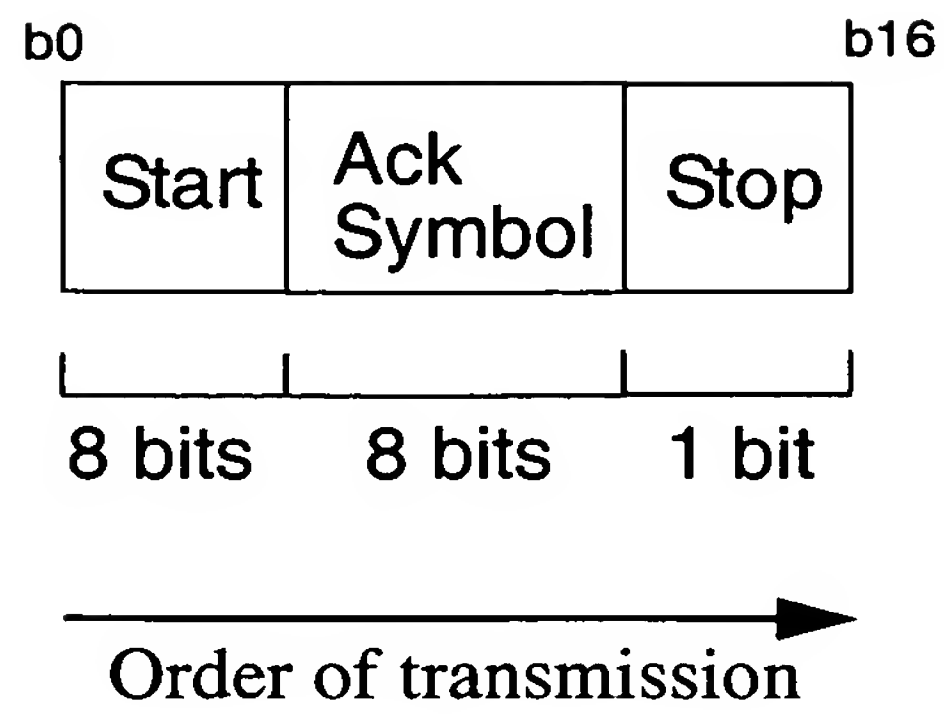
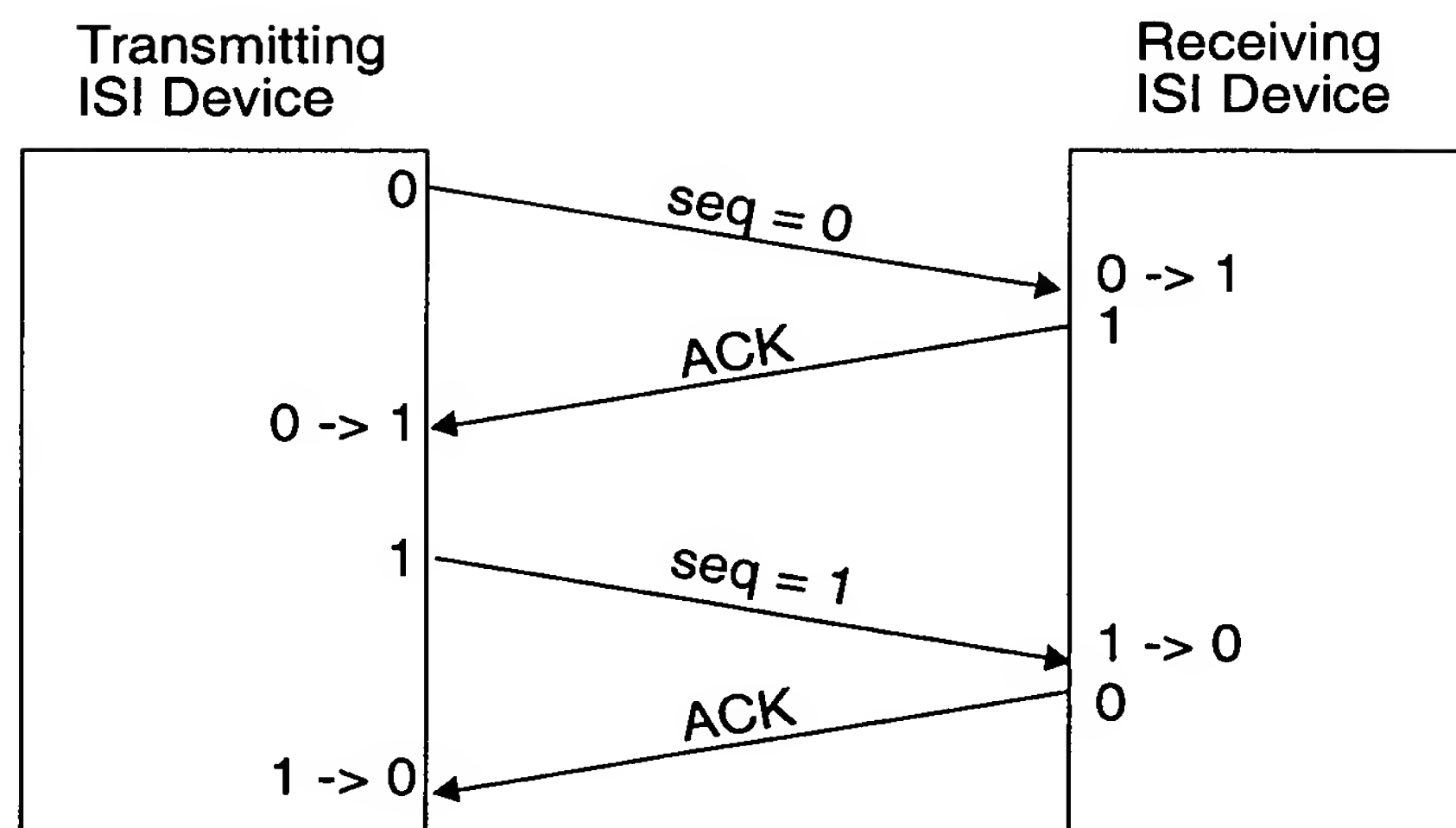


FIG. 34

*FIG. 35**FIG. 36*

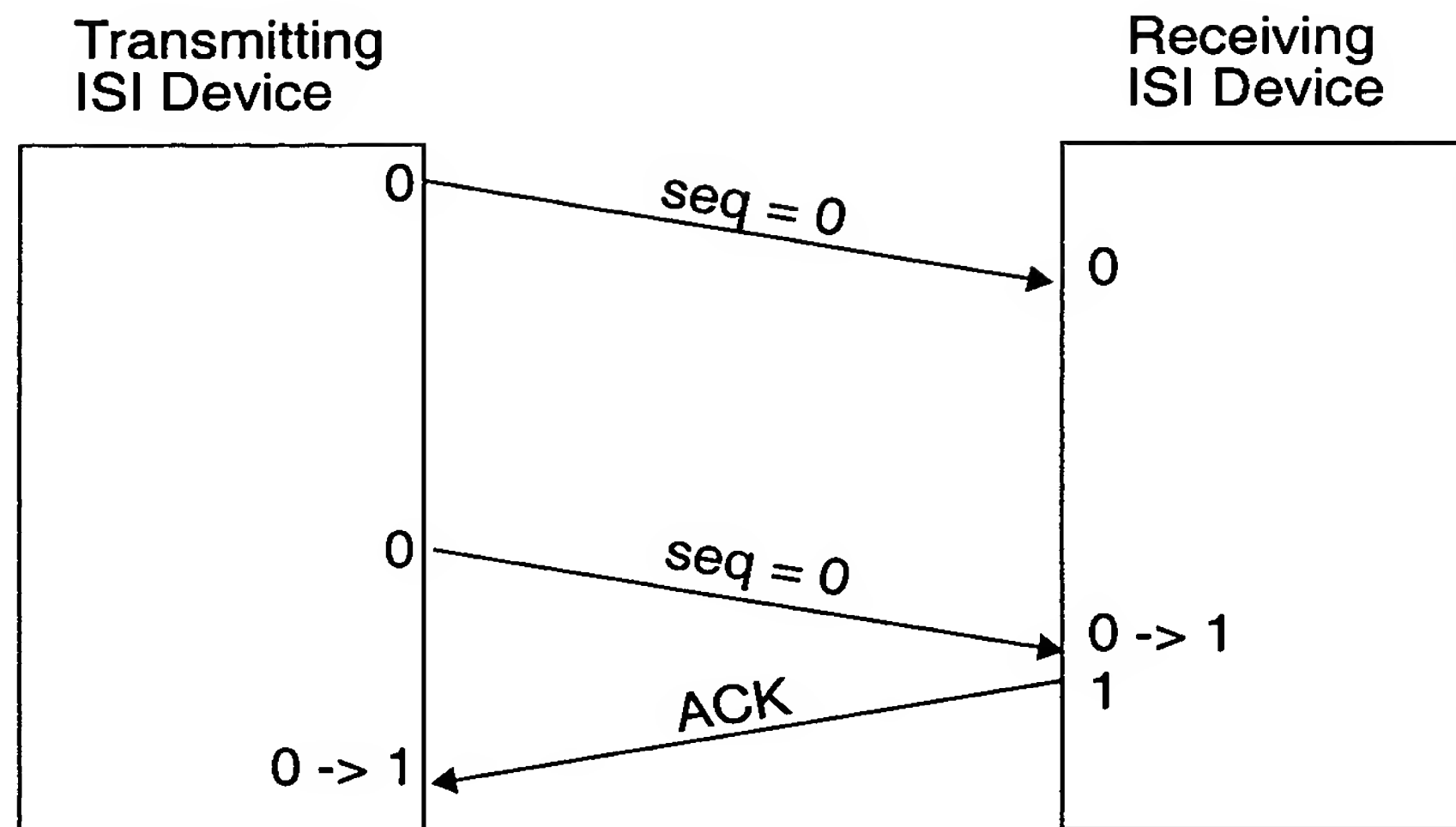


FIG. 37

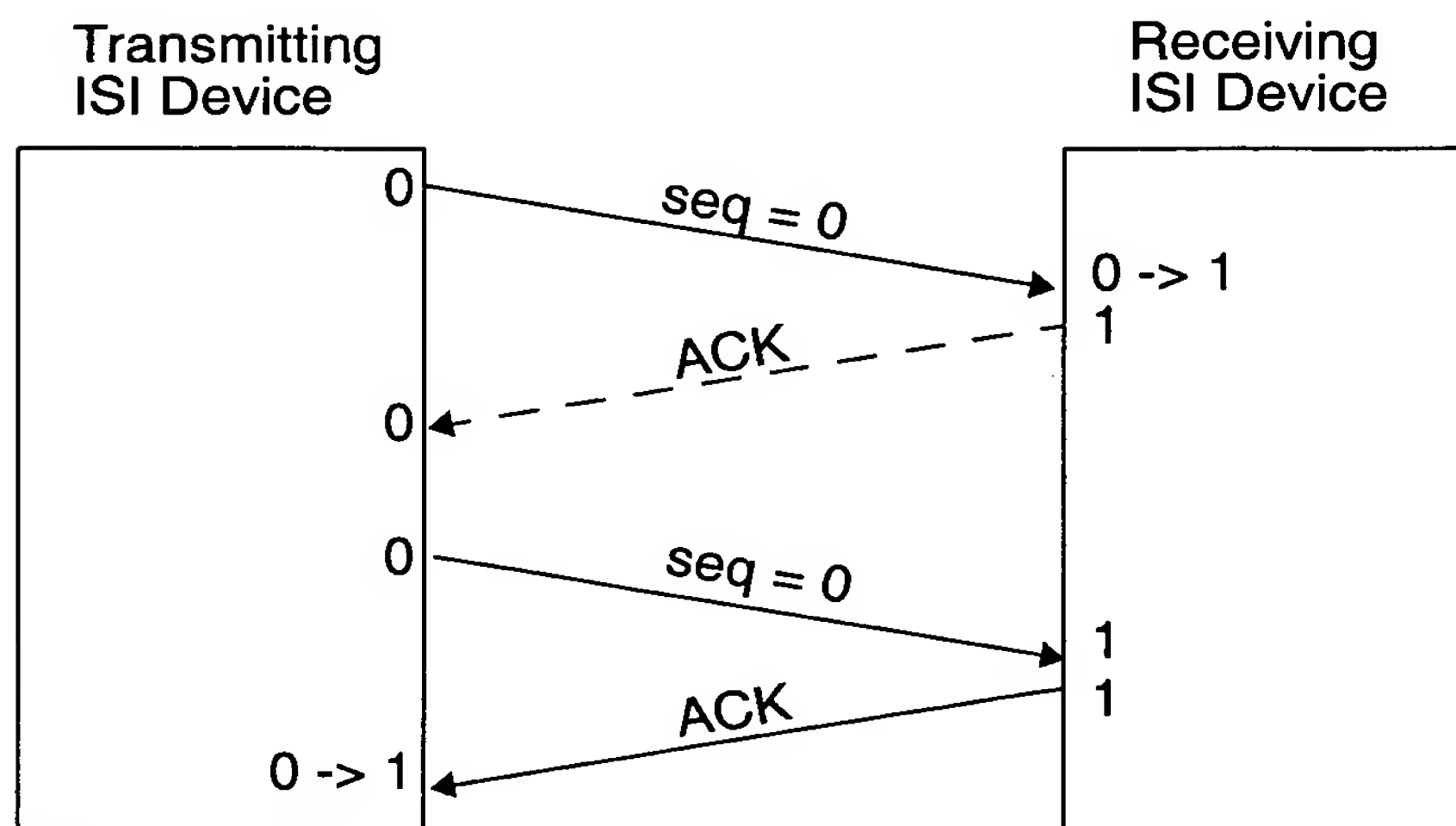


FIG. 38

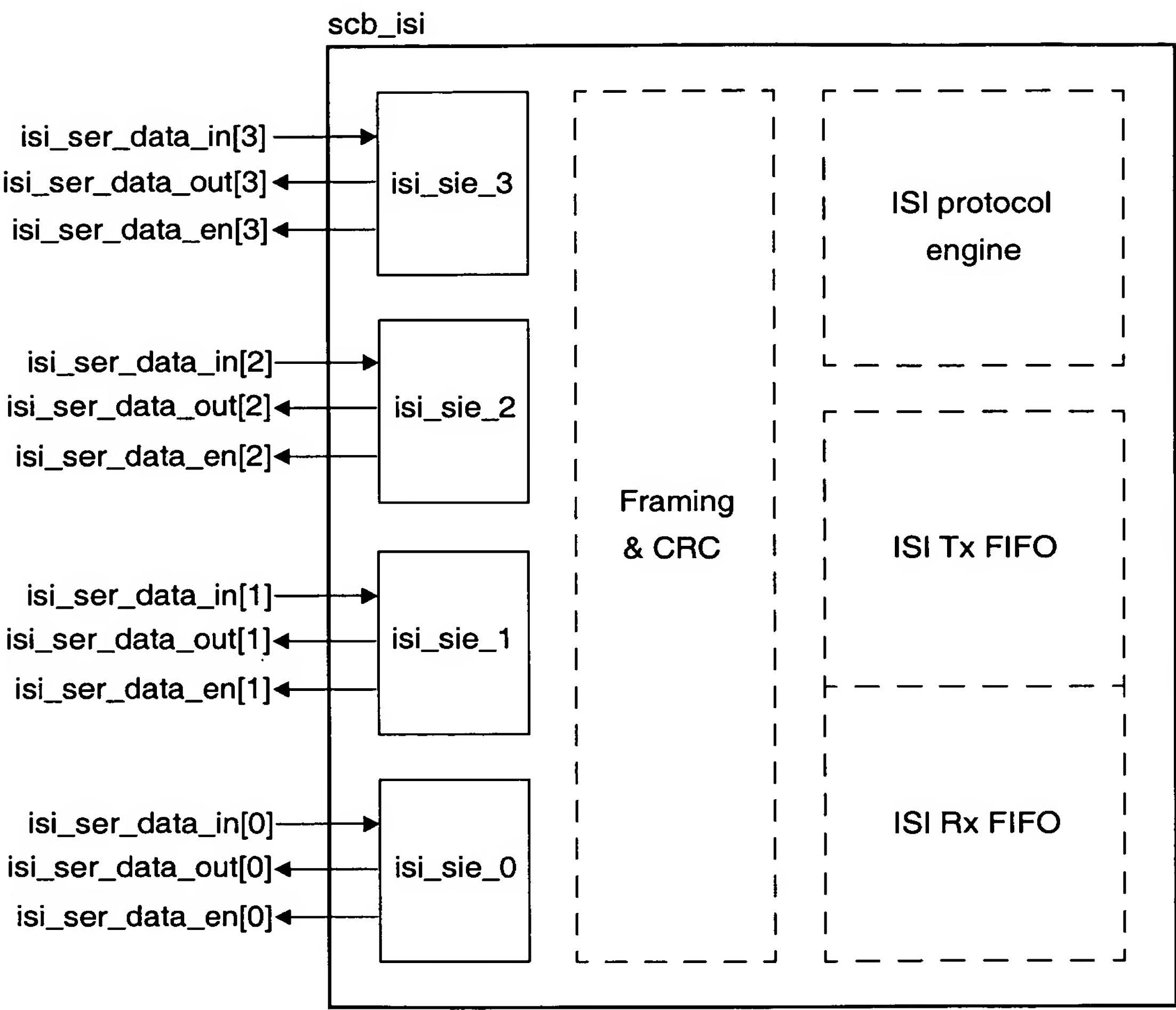


FIG. 39

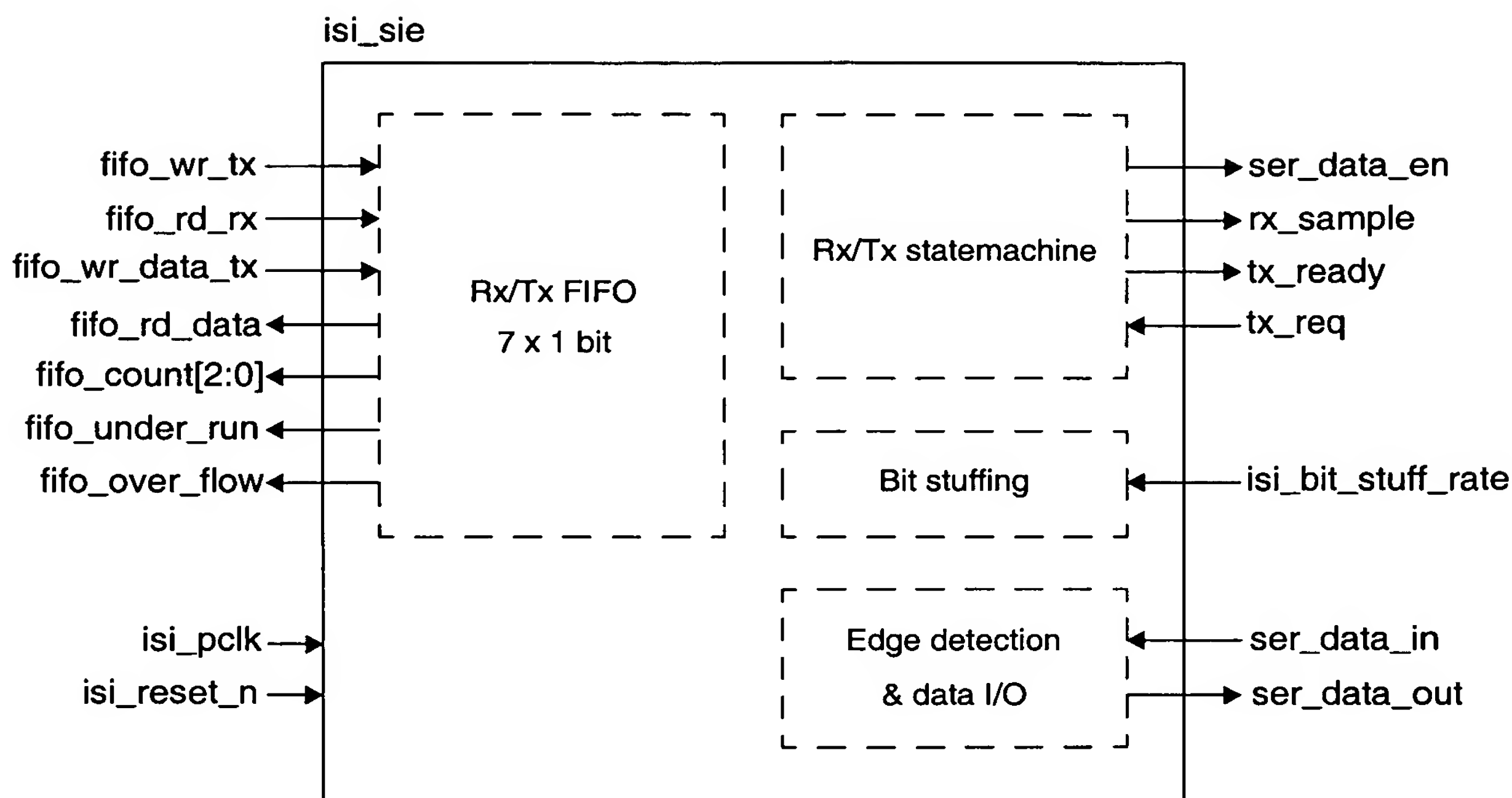


FIG. 40

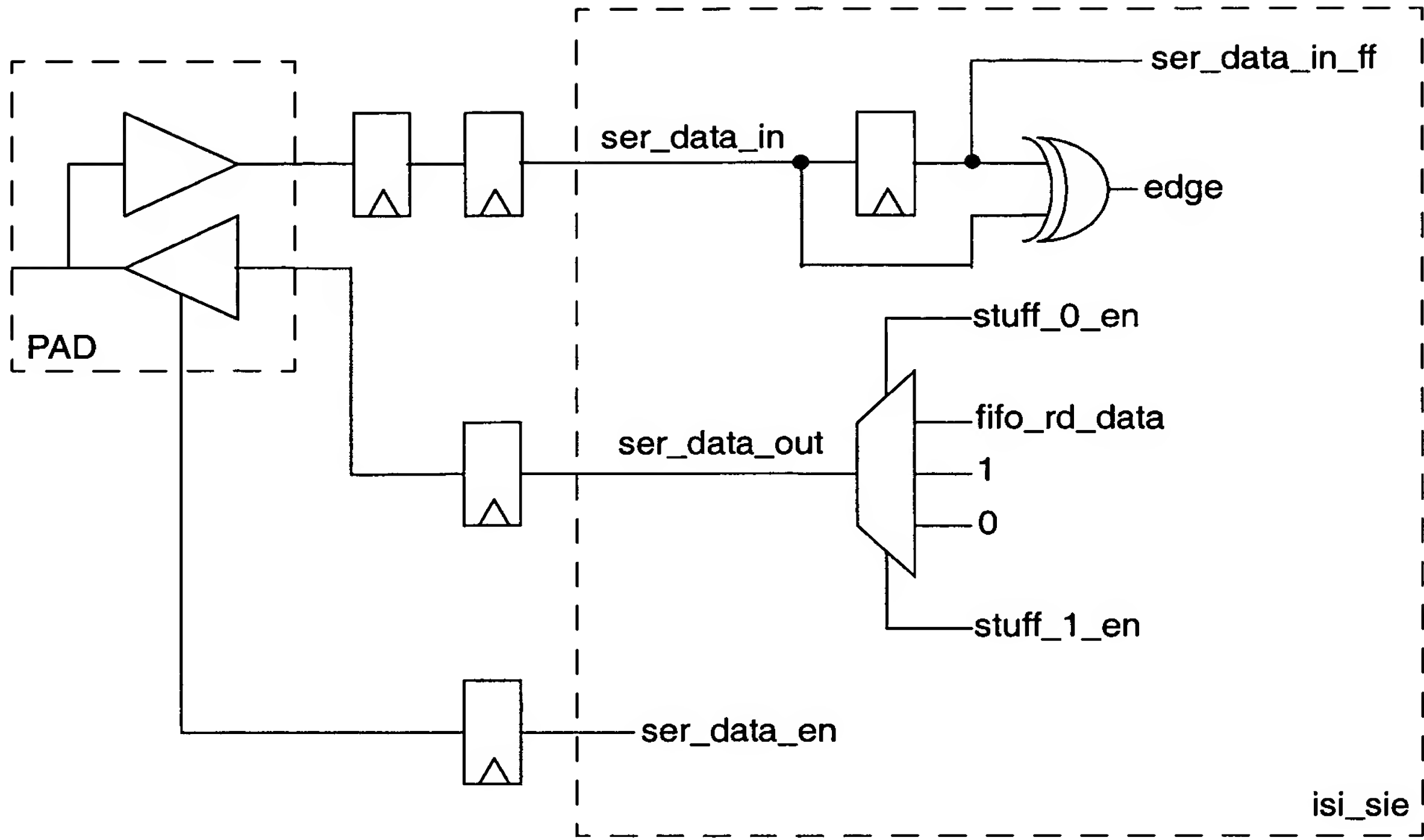


FIG. 41

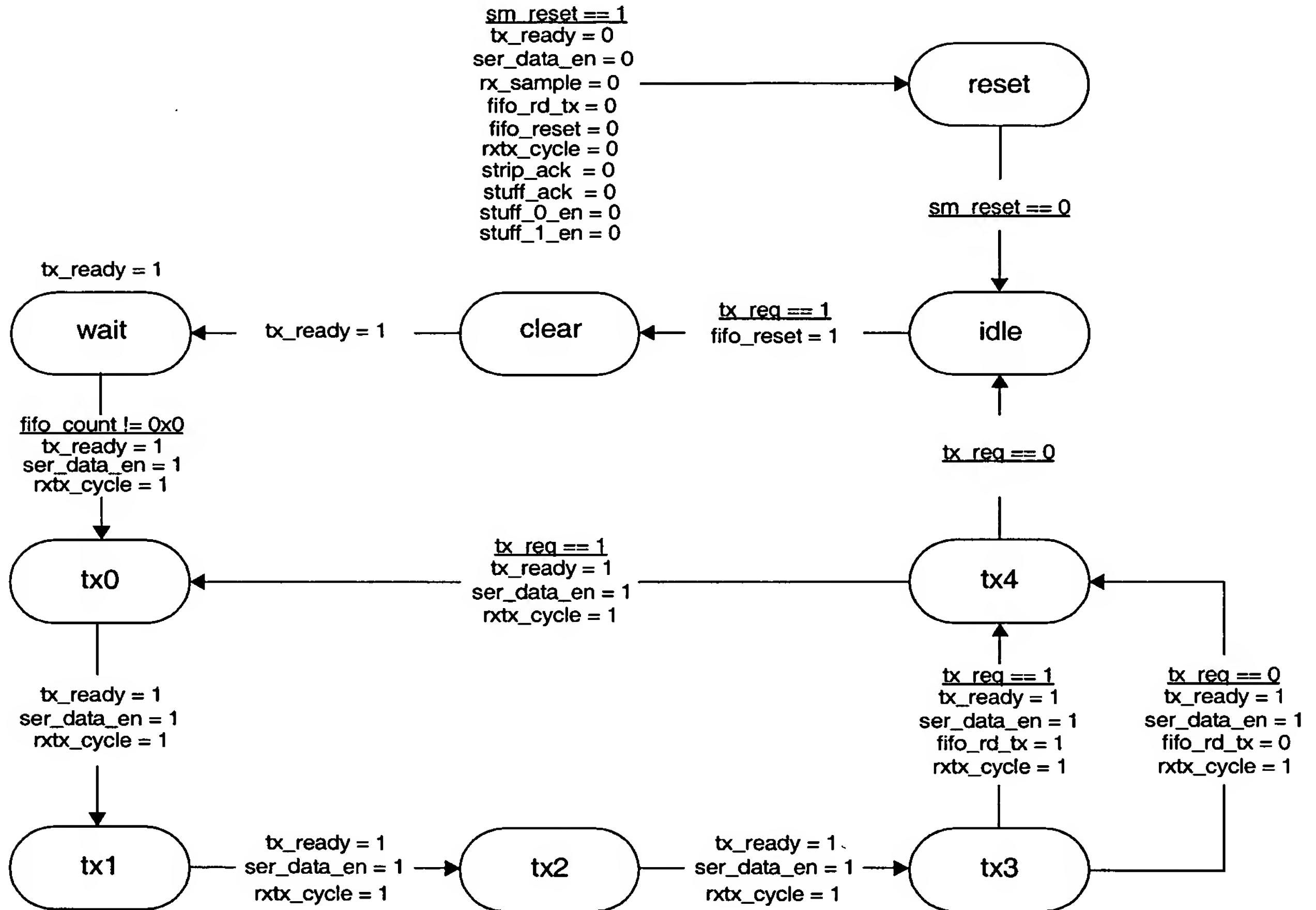


FIG. 42

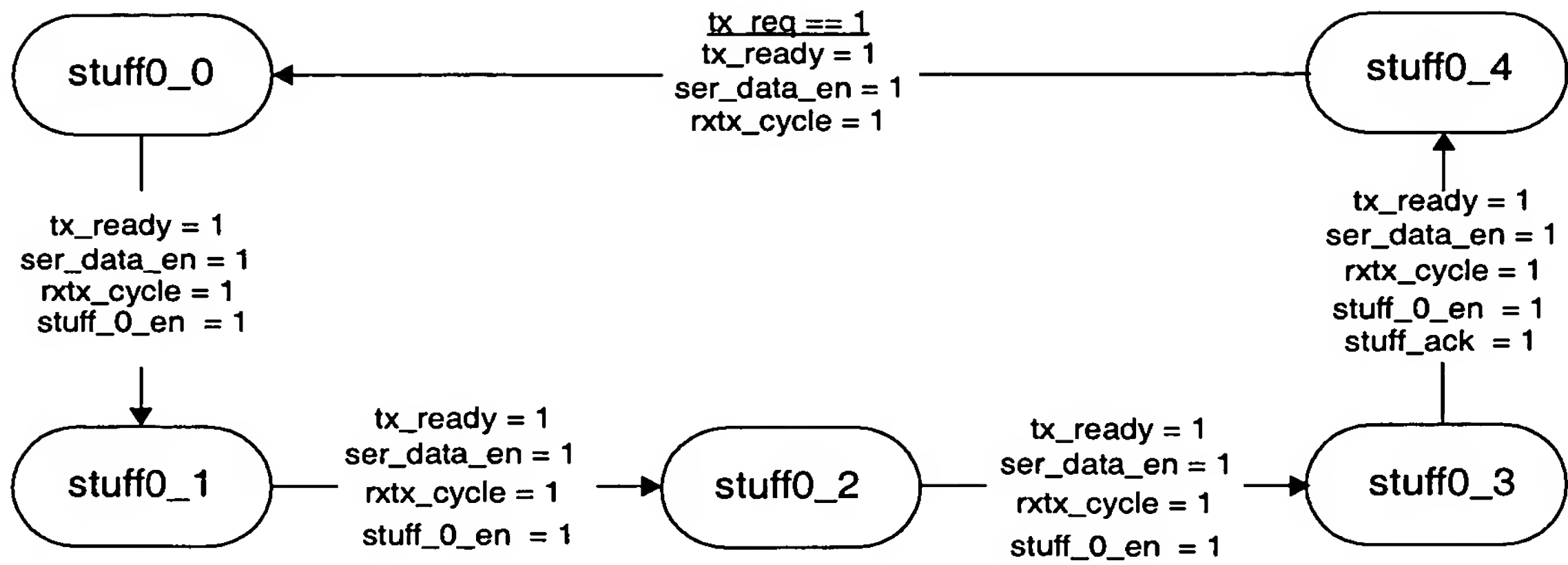


FIG. 43

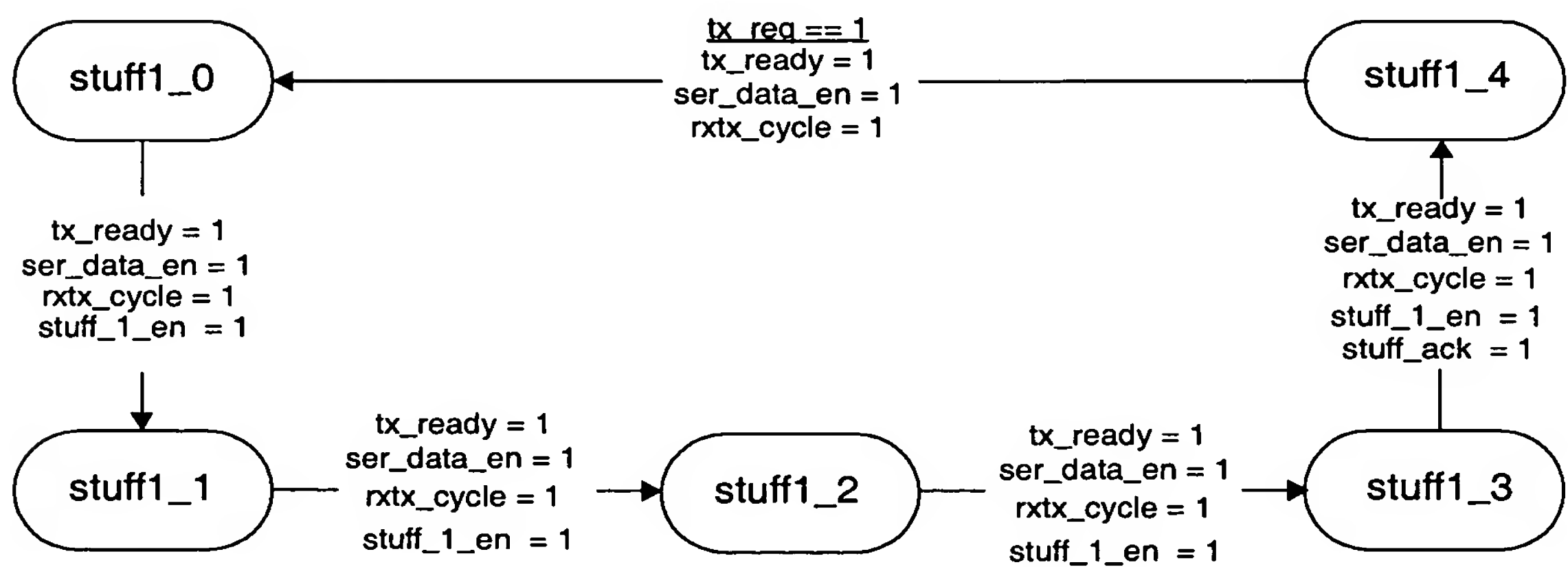


FIG. 44

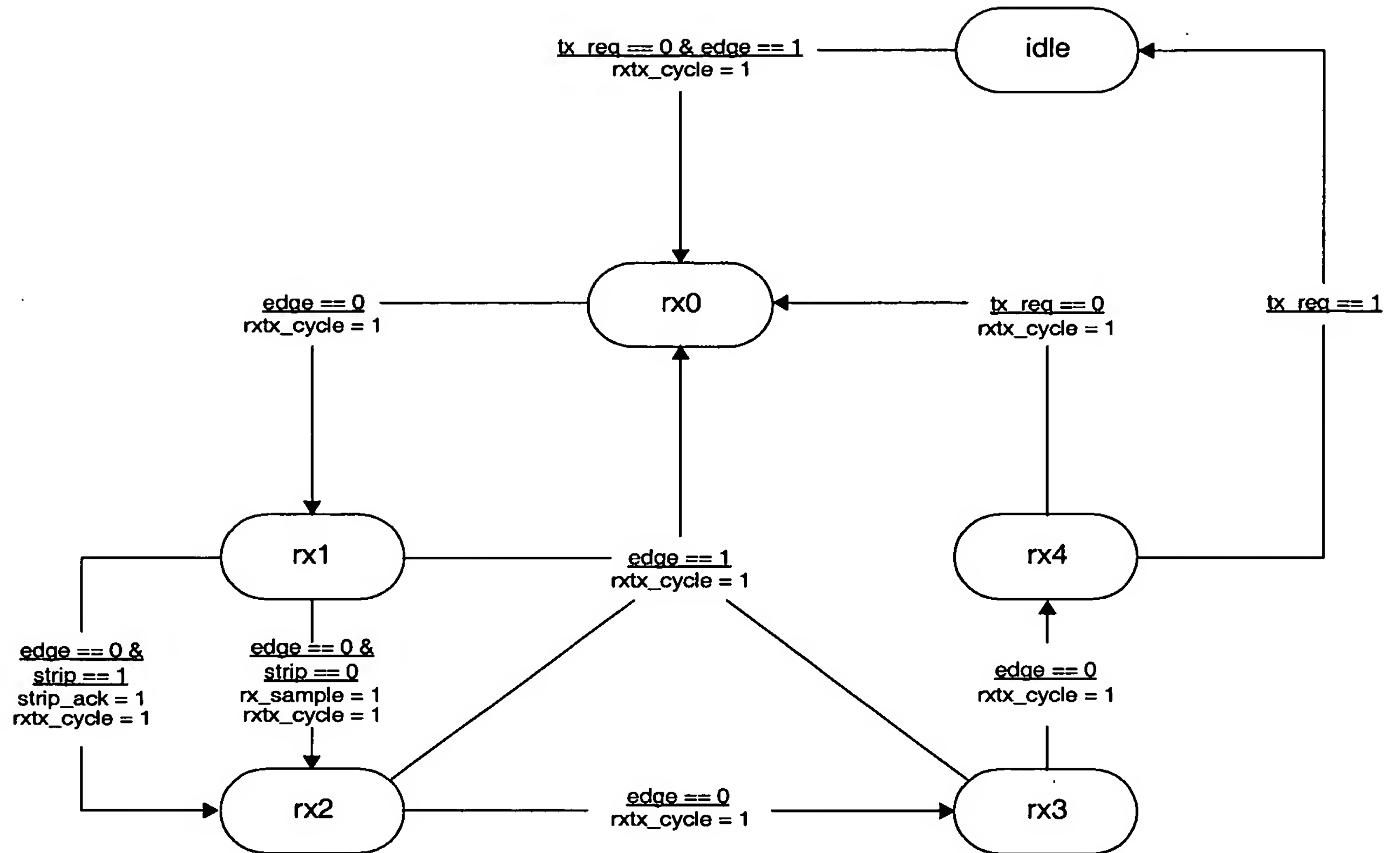


FIG. 45

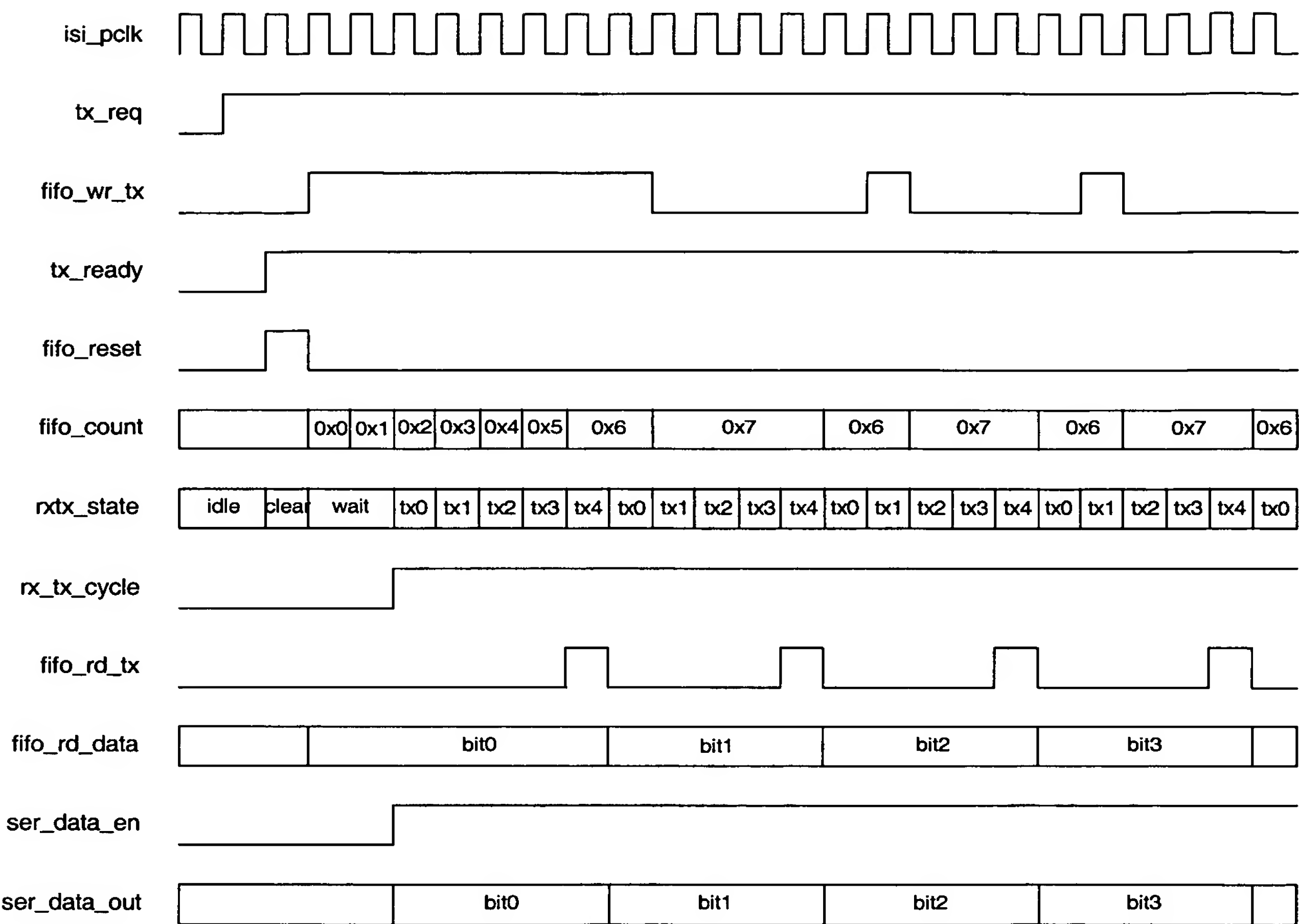


FIG. 46

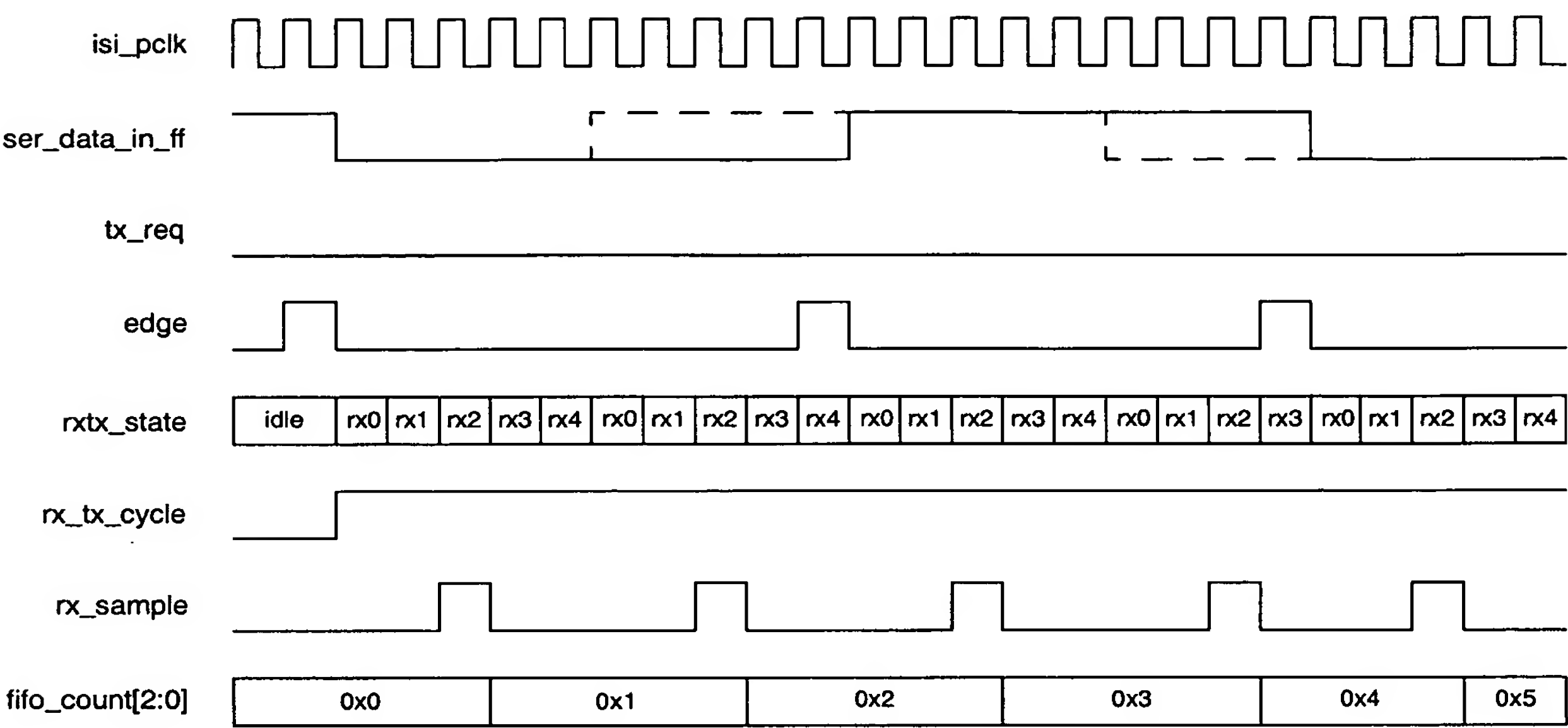


FIG. 47

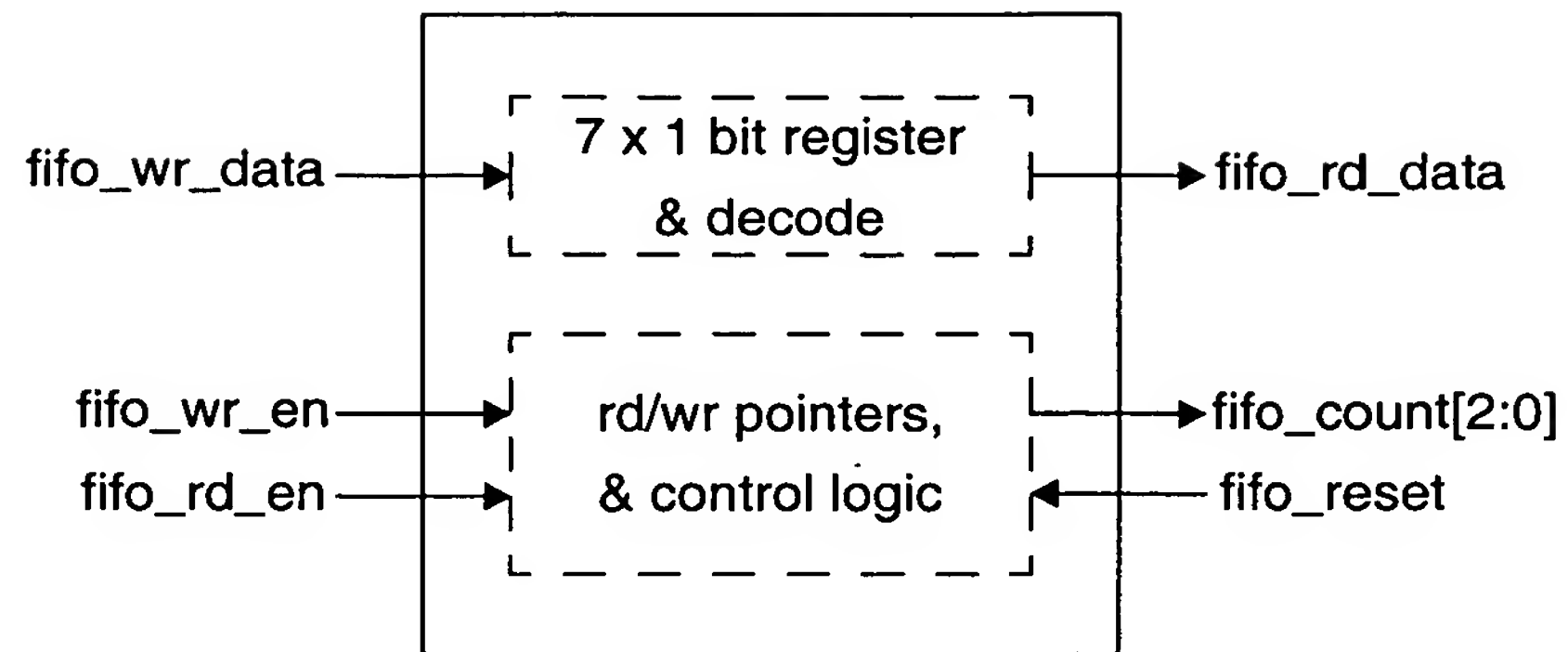


FIG. 48

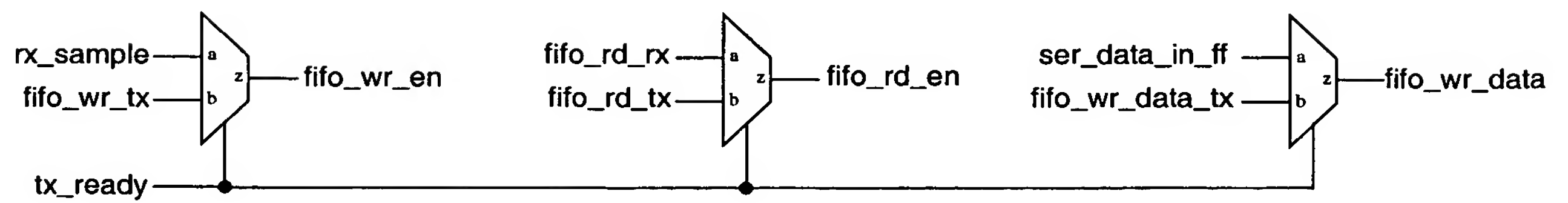


FIG. 49

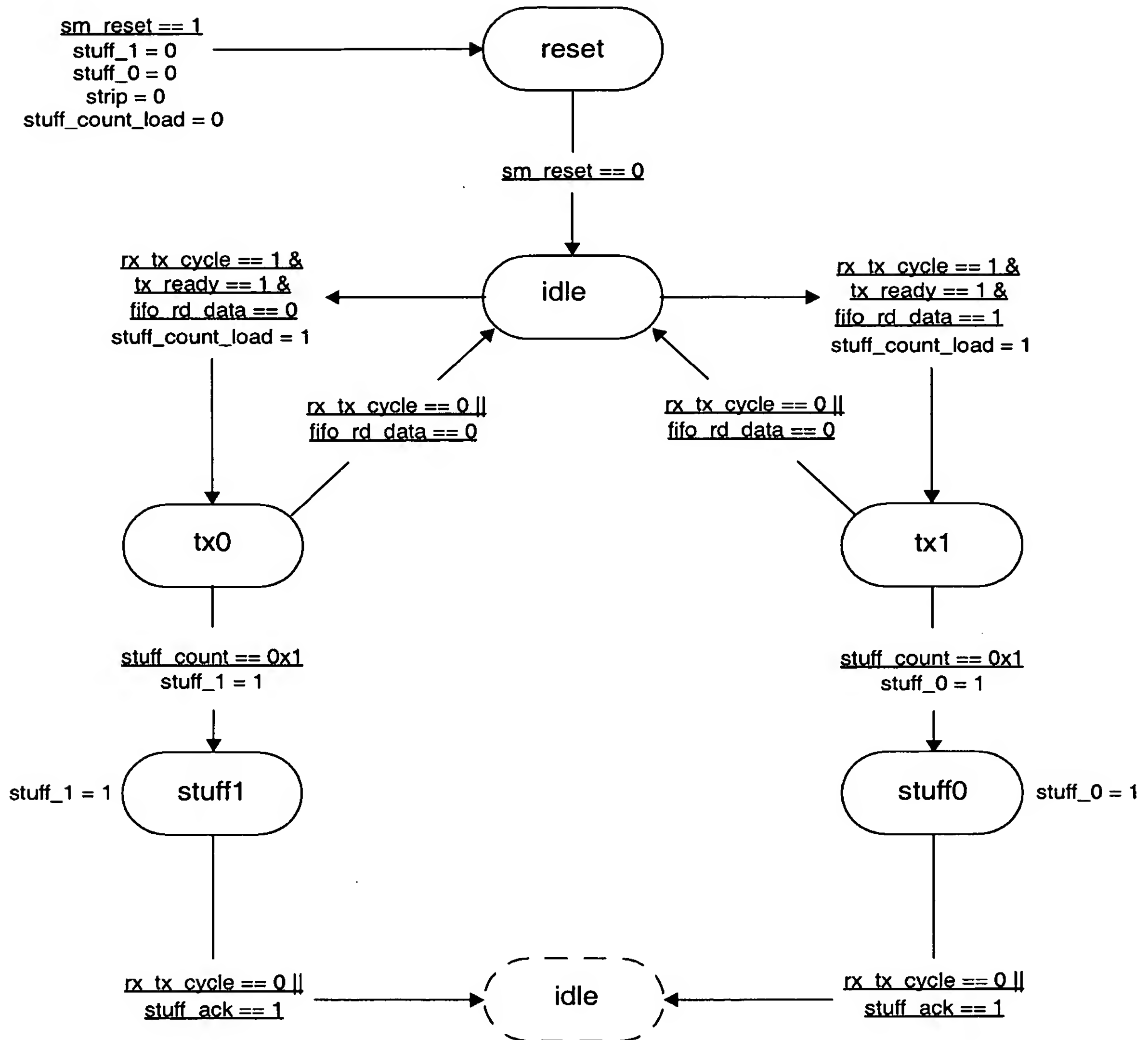


FIG. 50

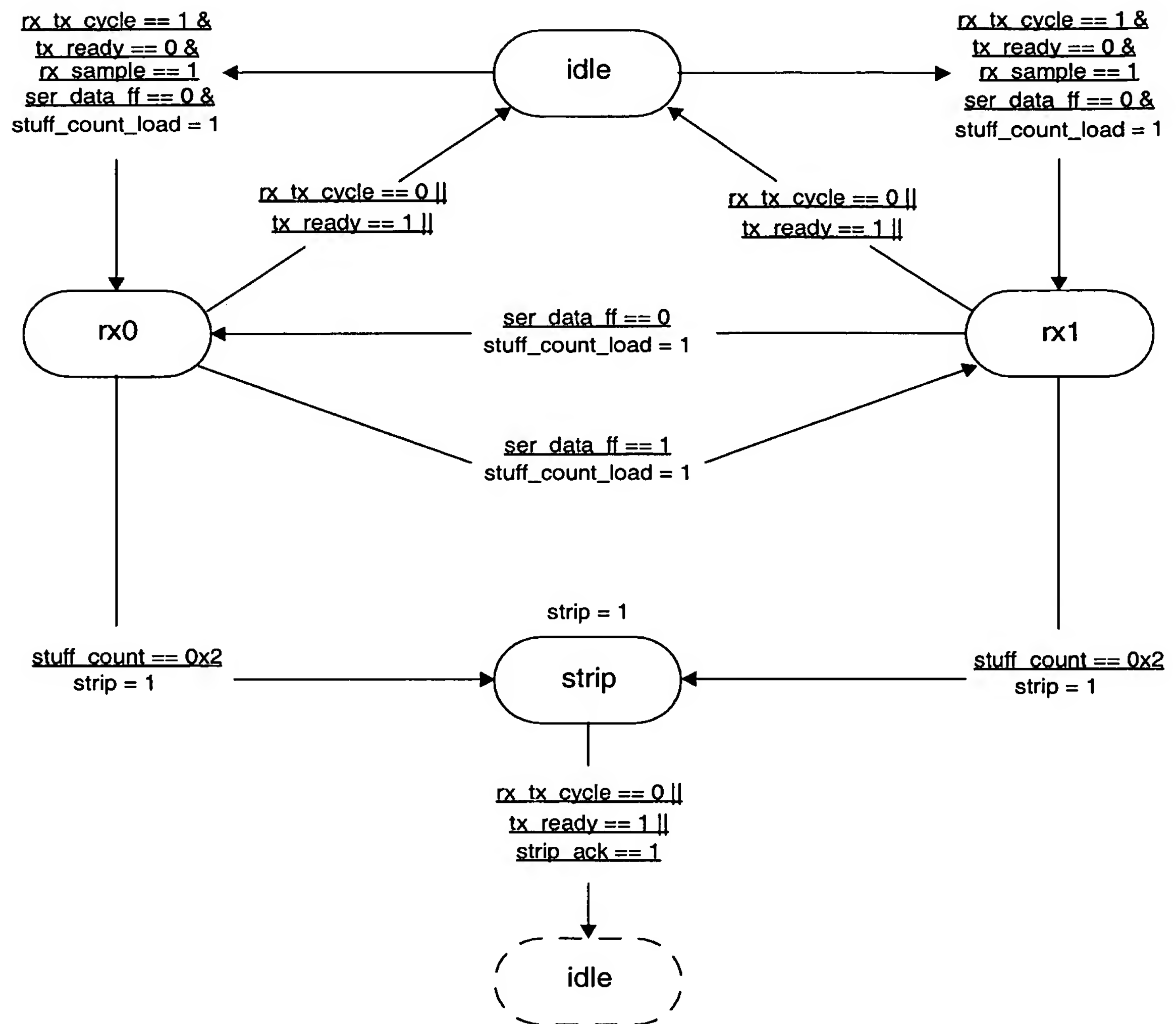


FIG. 51

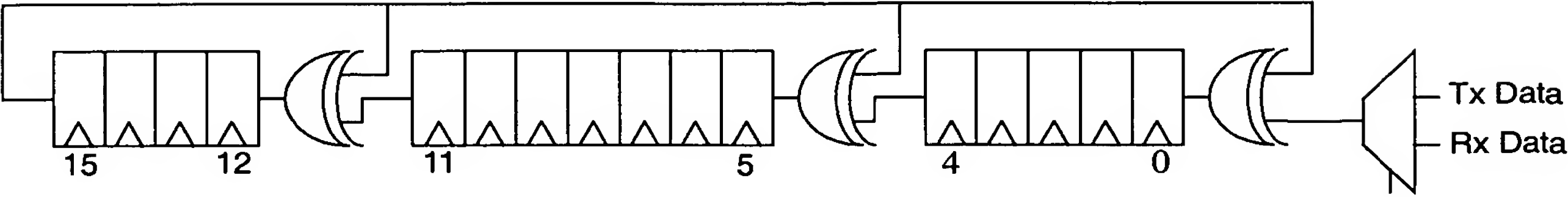


FIG. 52

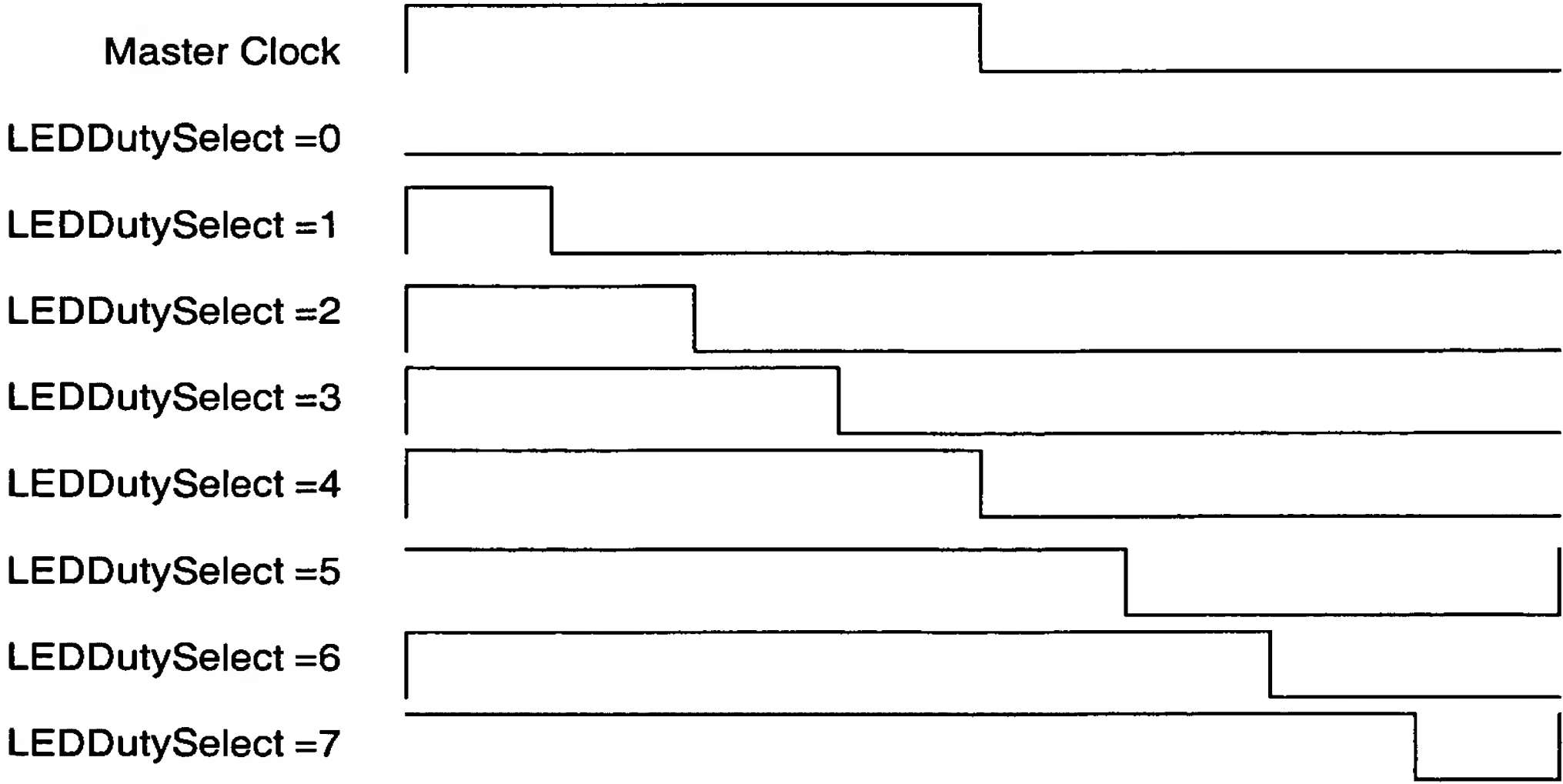


FIG. 54

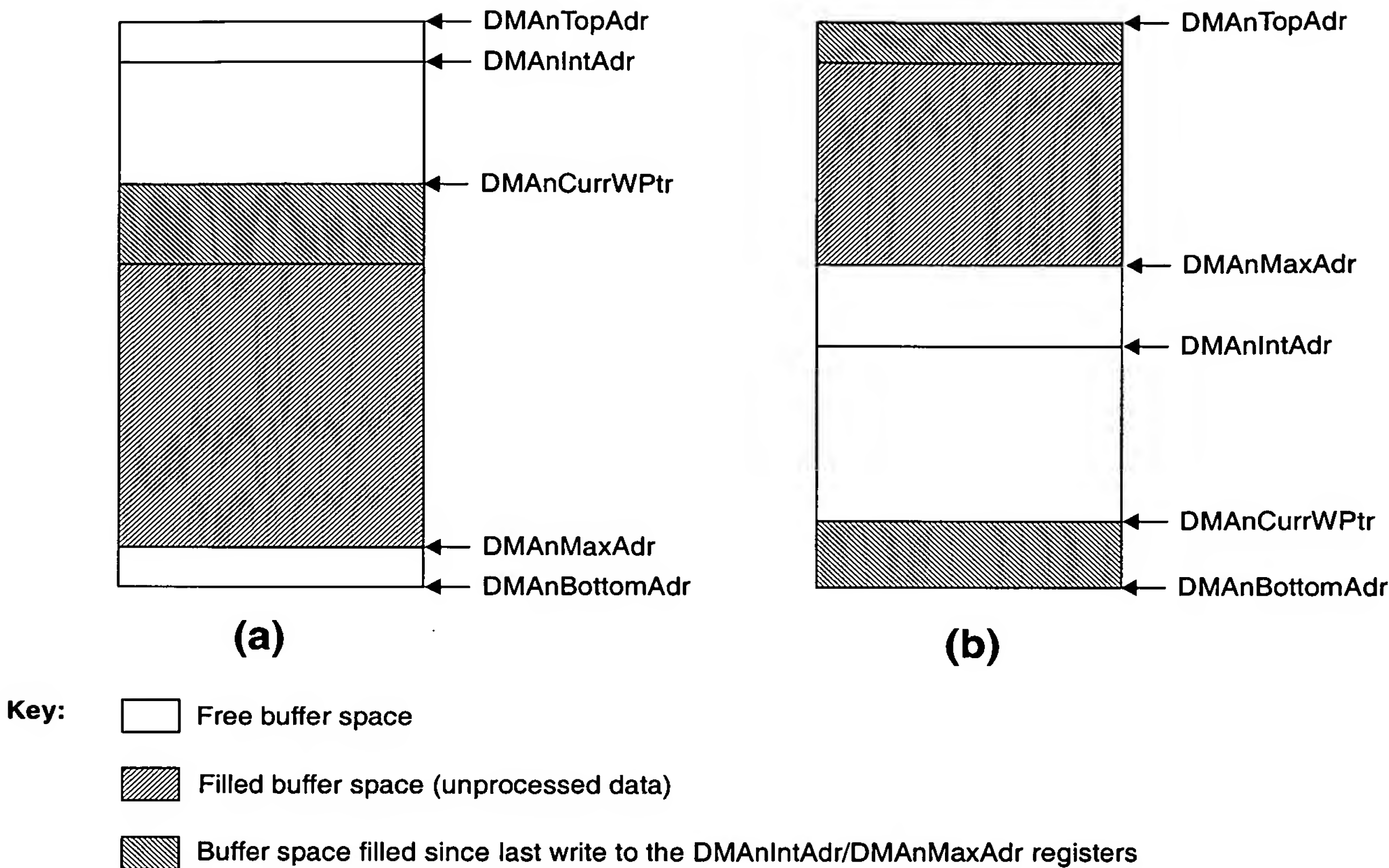


FIG. 53

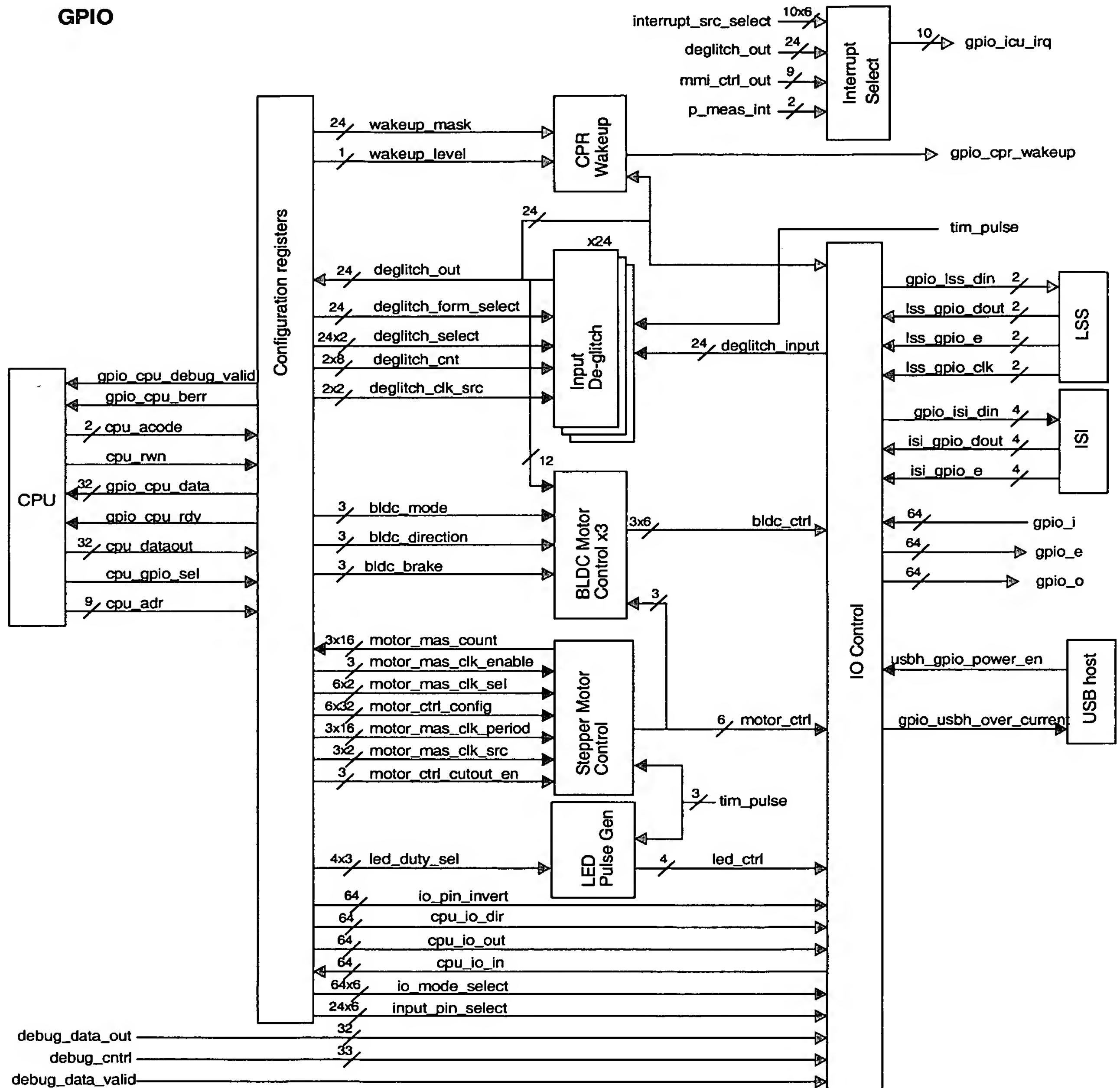


FIG. 55

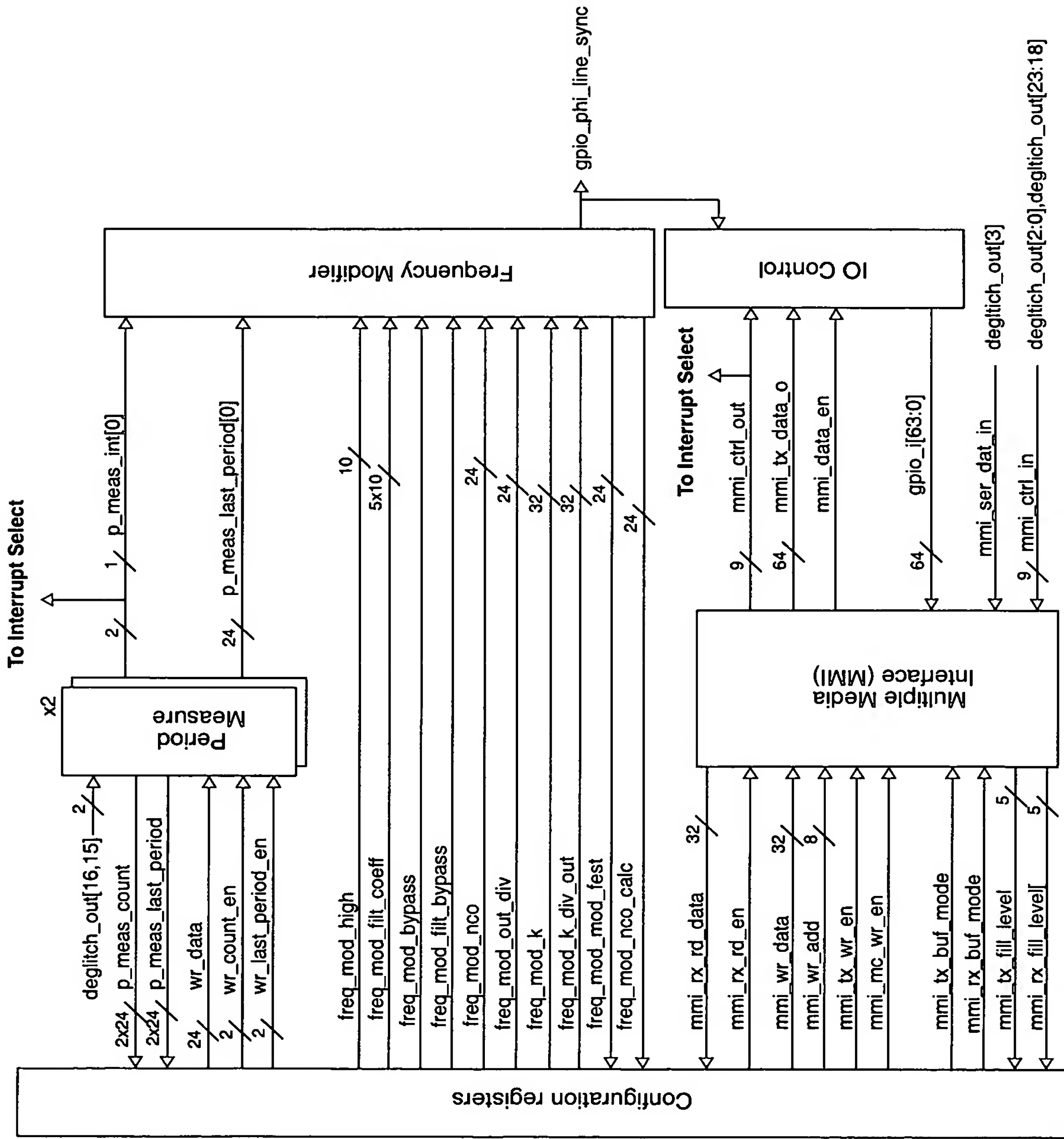


FIG. 56



FIG. 57

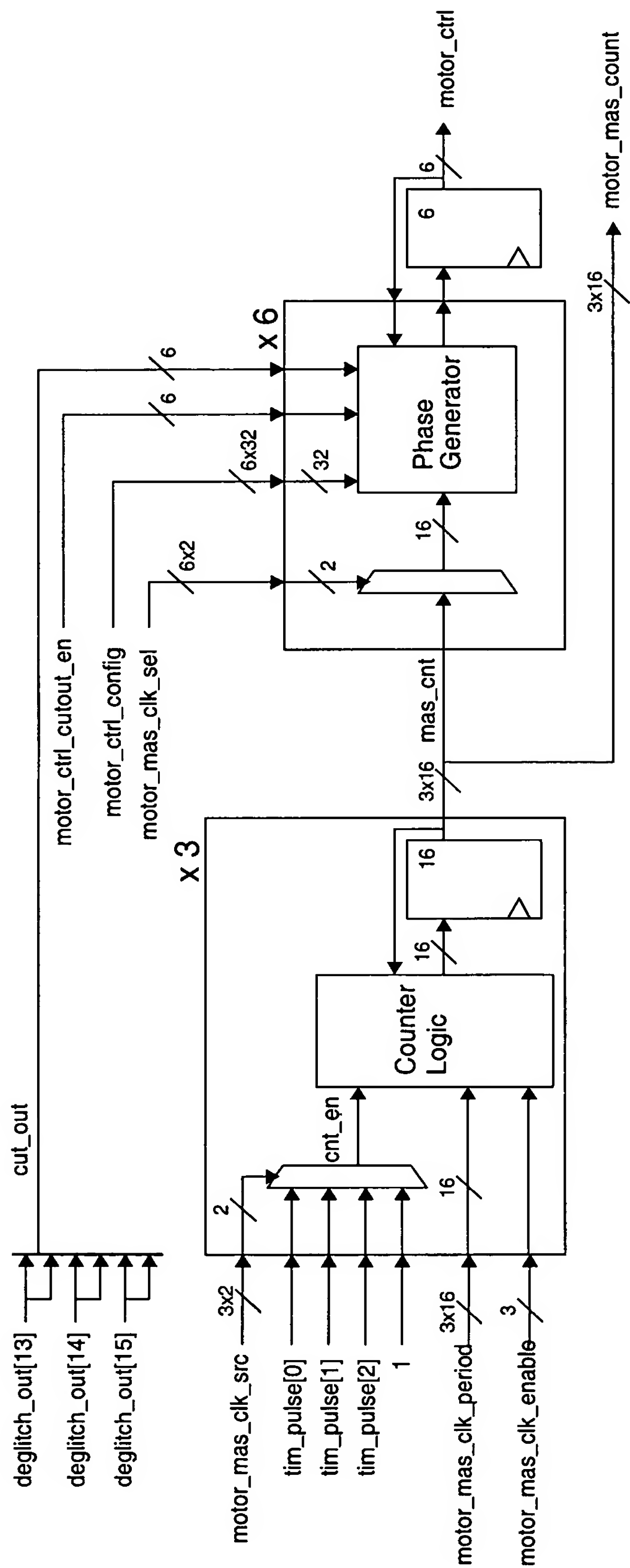


FIG. 58

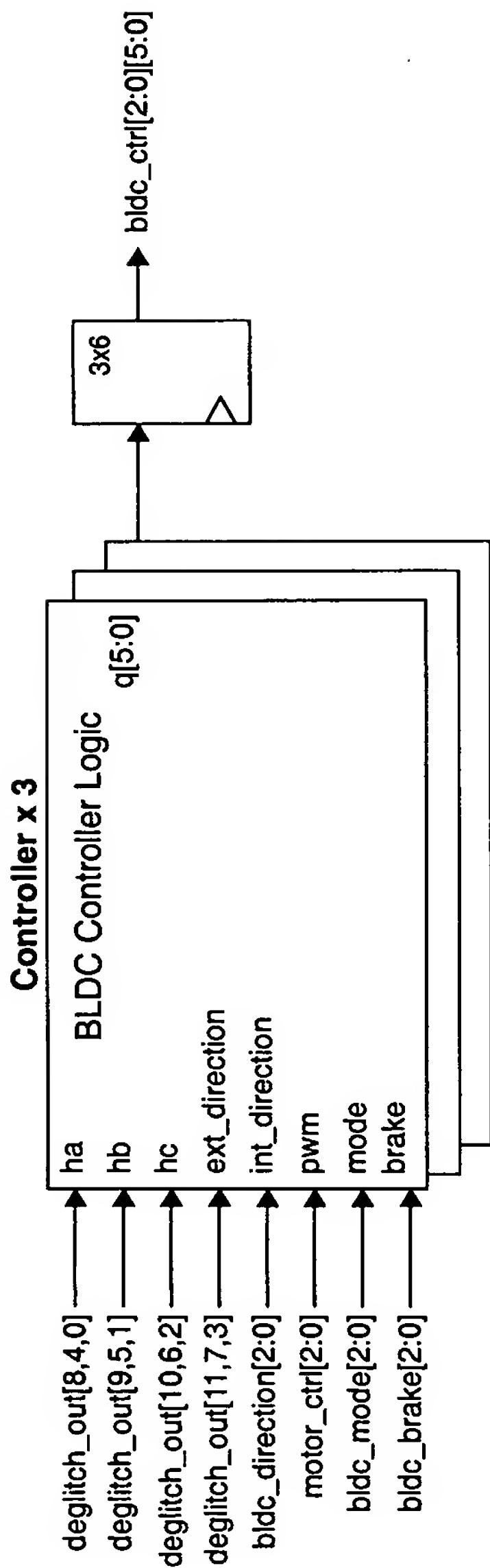


FIG. 59

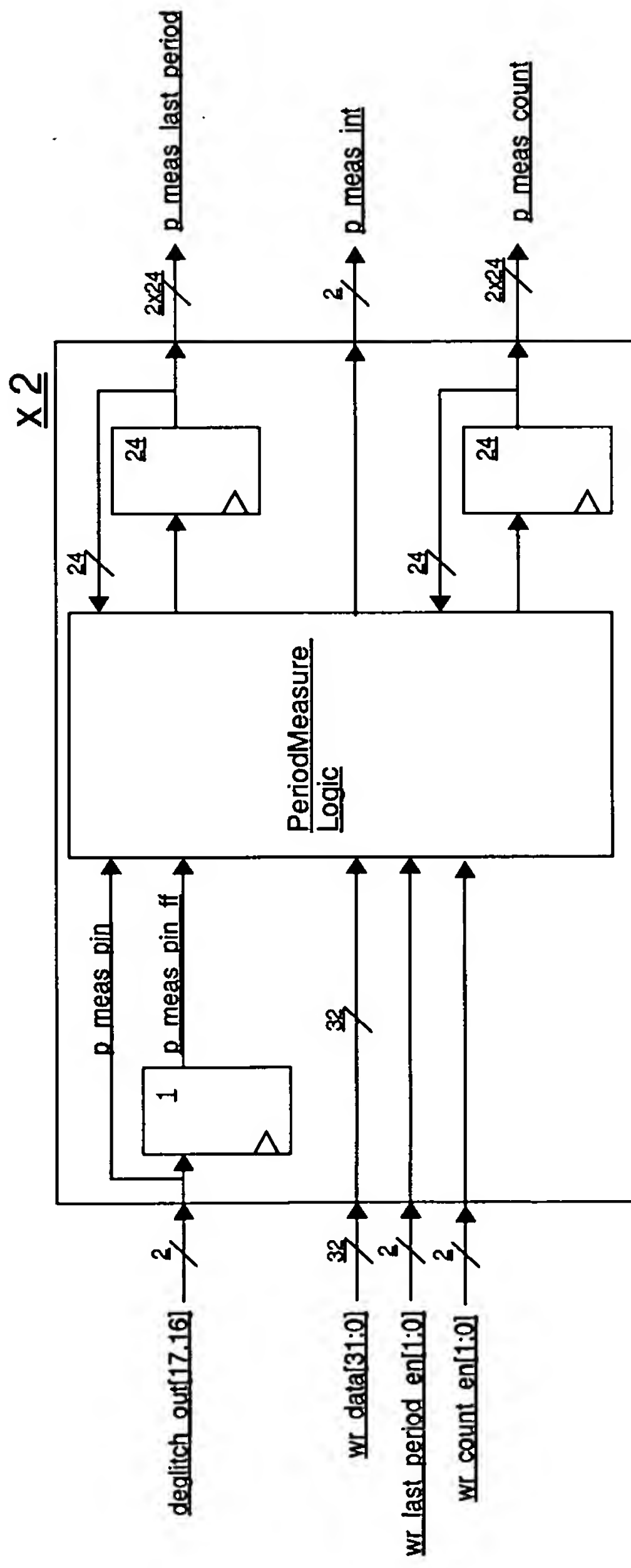


FIG. 60

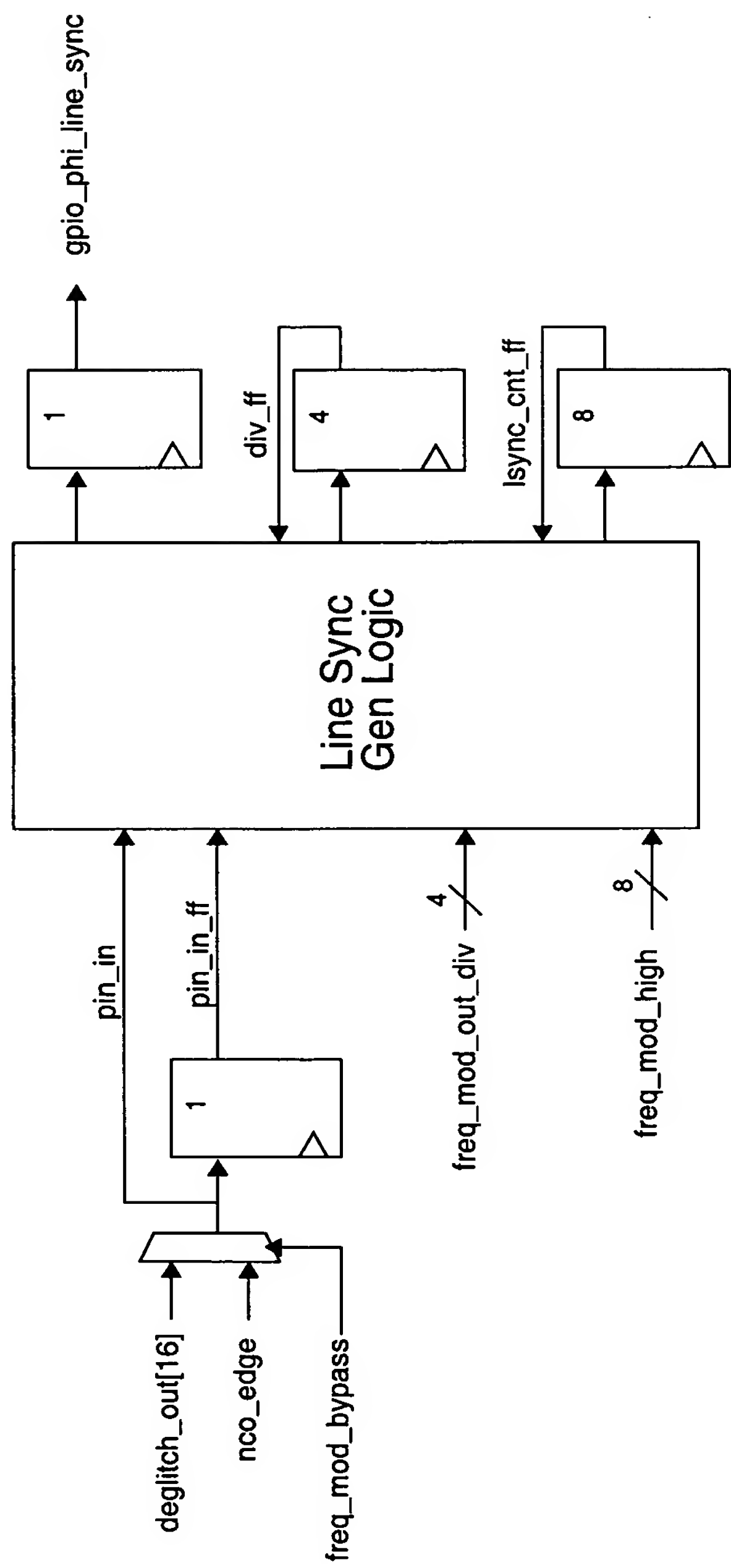


FIG. 61

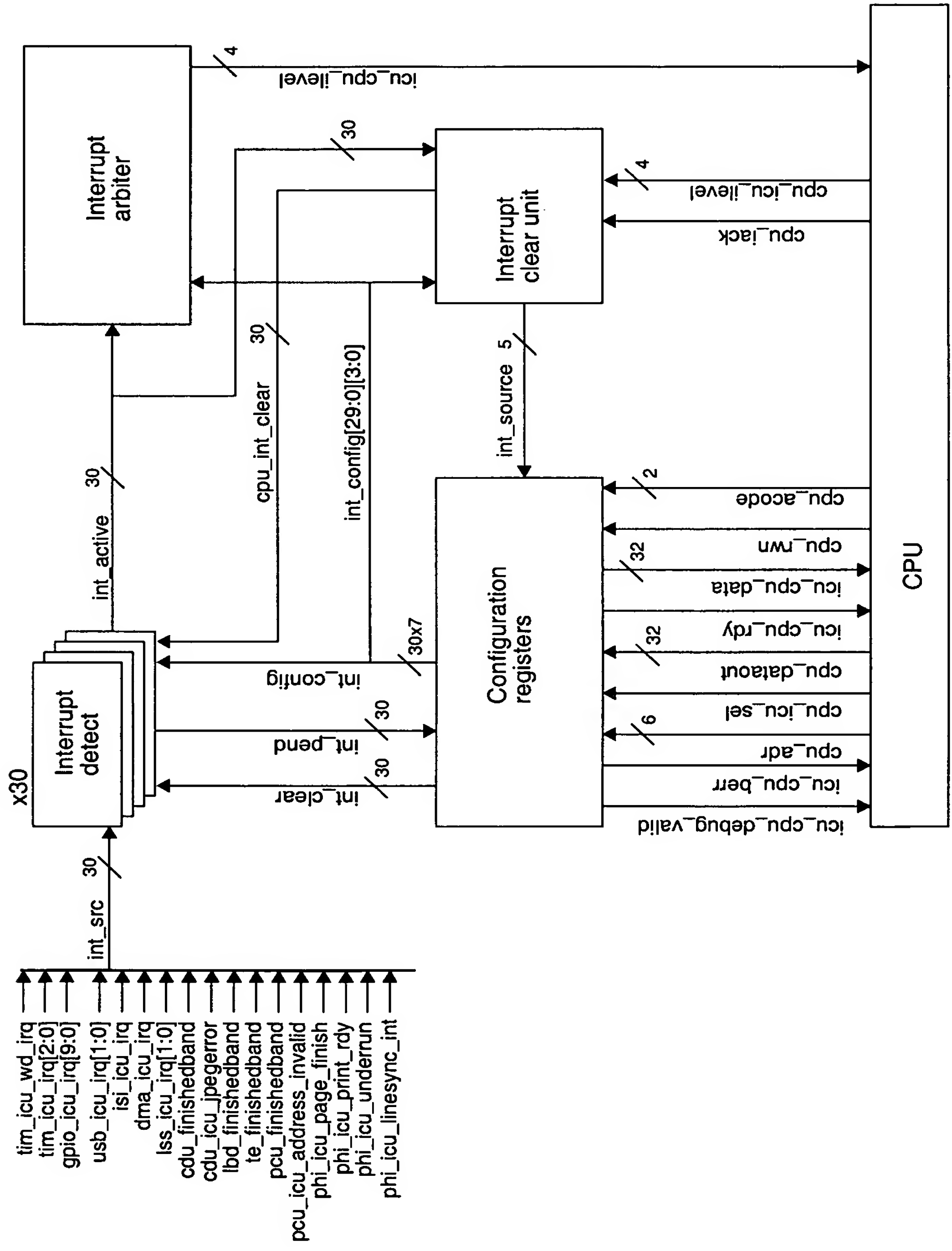
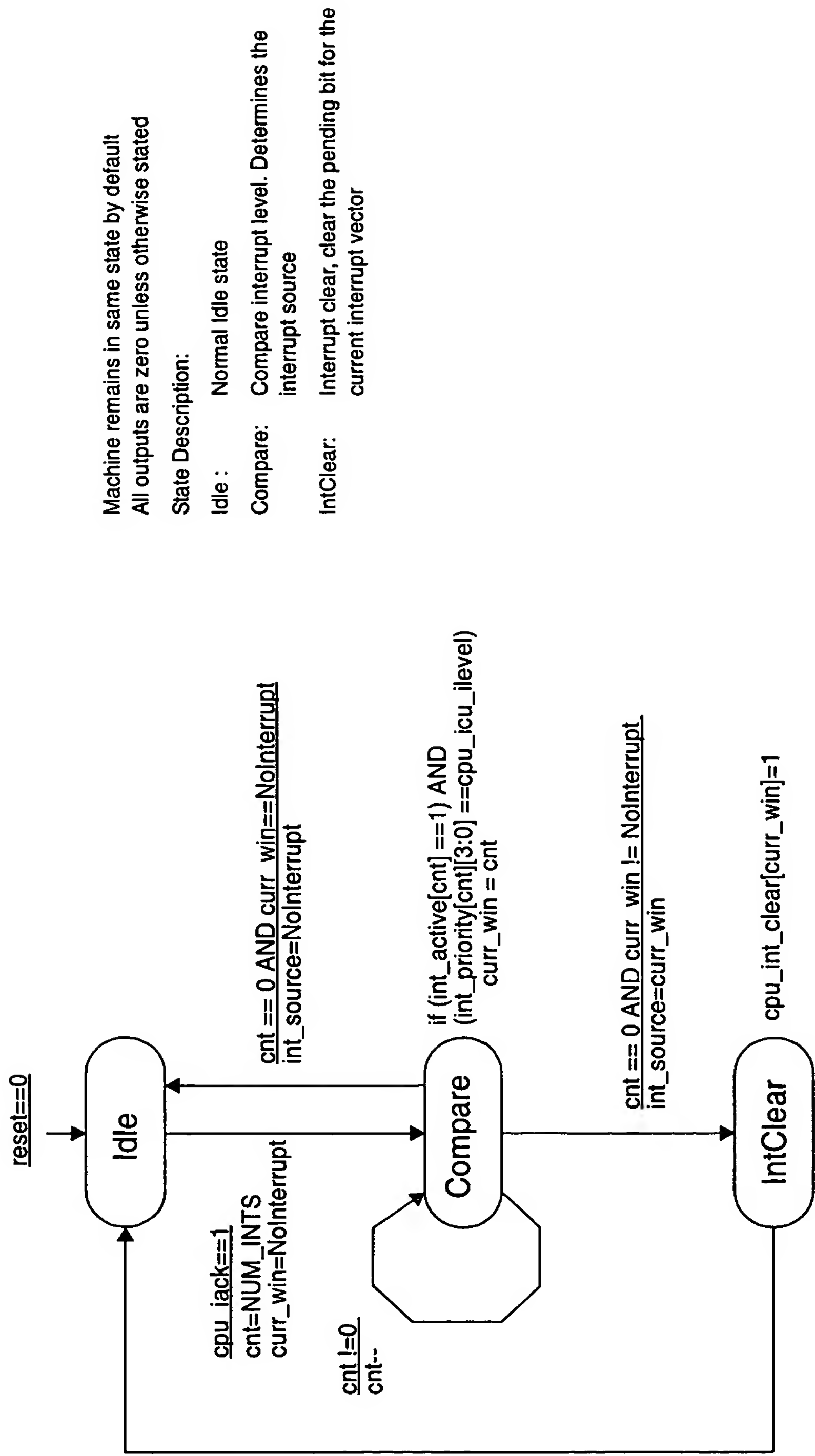


FIG. 62



Machine remains in same state by default
All outputs are zero unless otherwise stated

State Description:

Idle :	Normal Idle state
Compare:	Compare interrupt level. Determines the interrupt source
IntClear:	Interrupt clear, clear the pending bit for the current interrupt vector

FIG. 63

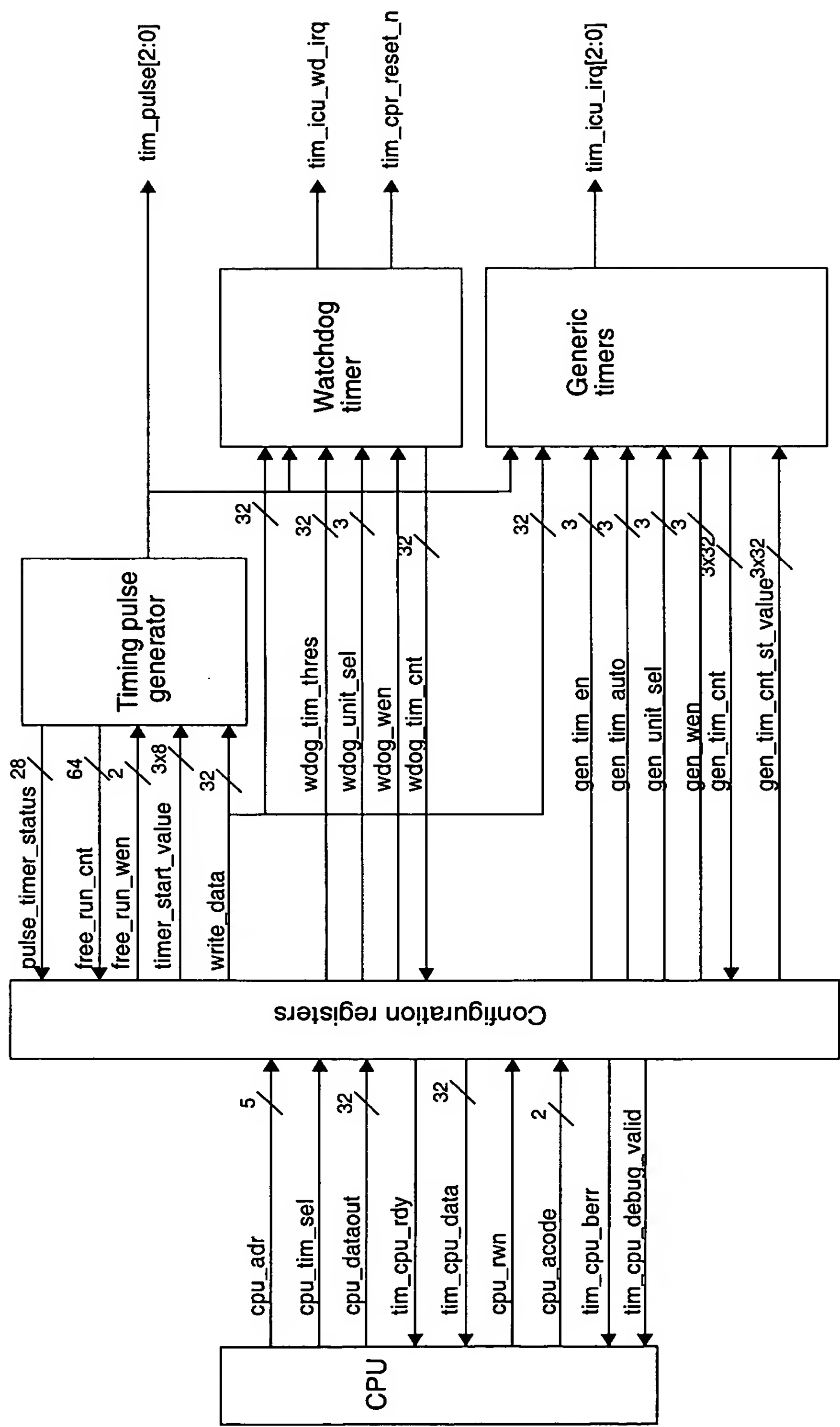


FIG. 63A

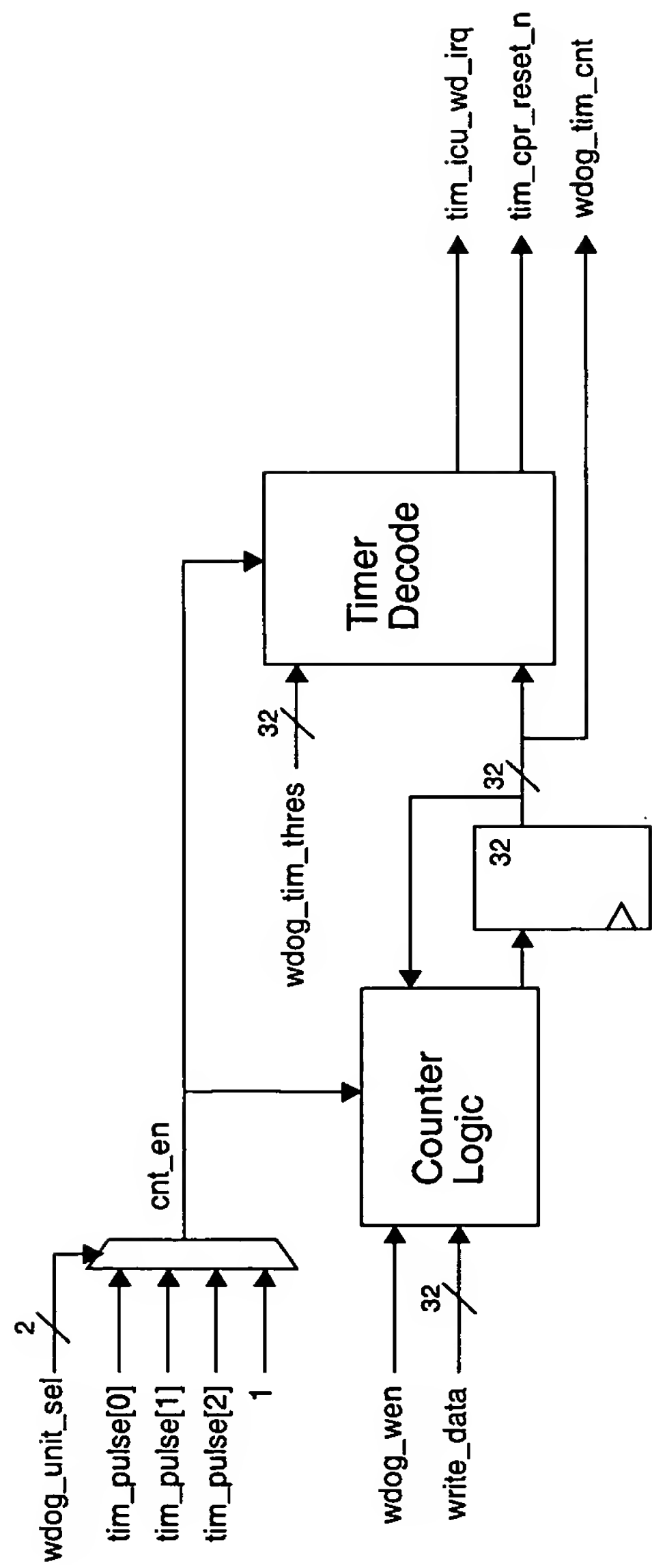


FIG. 64

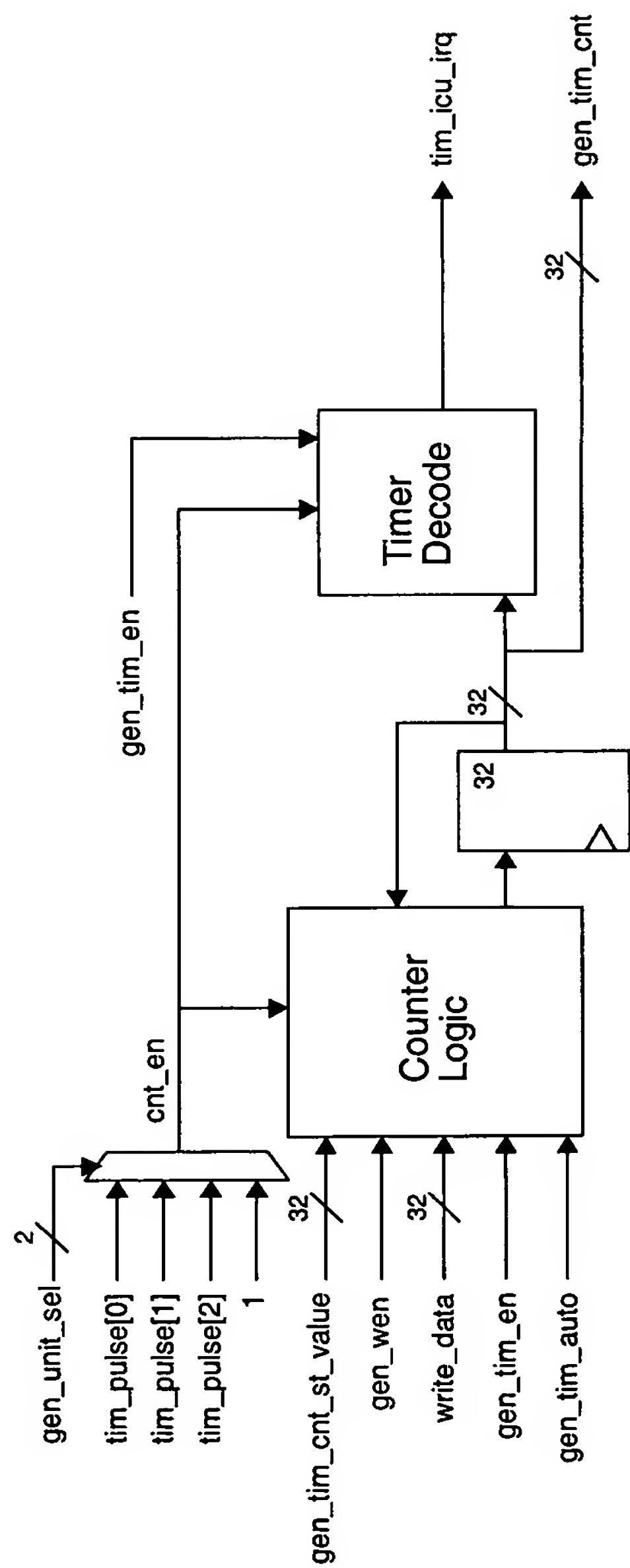


FIG. 65

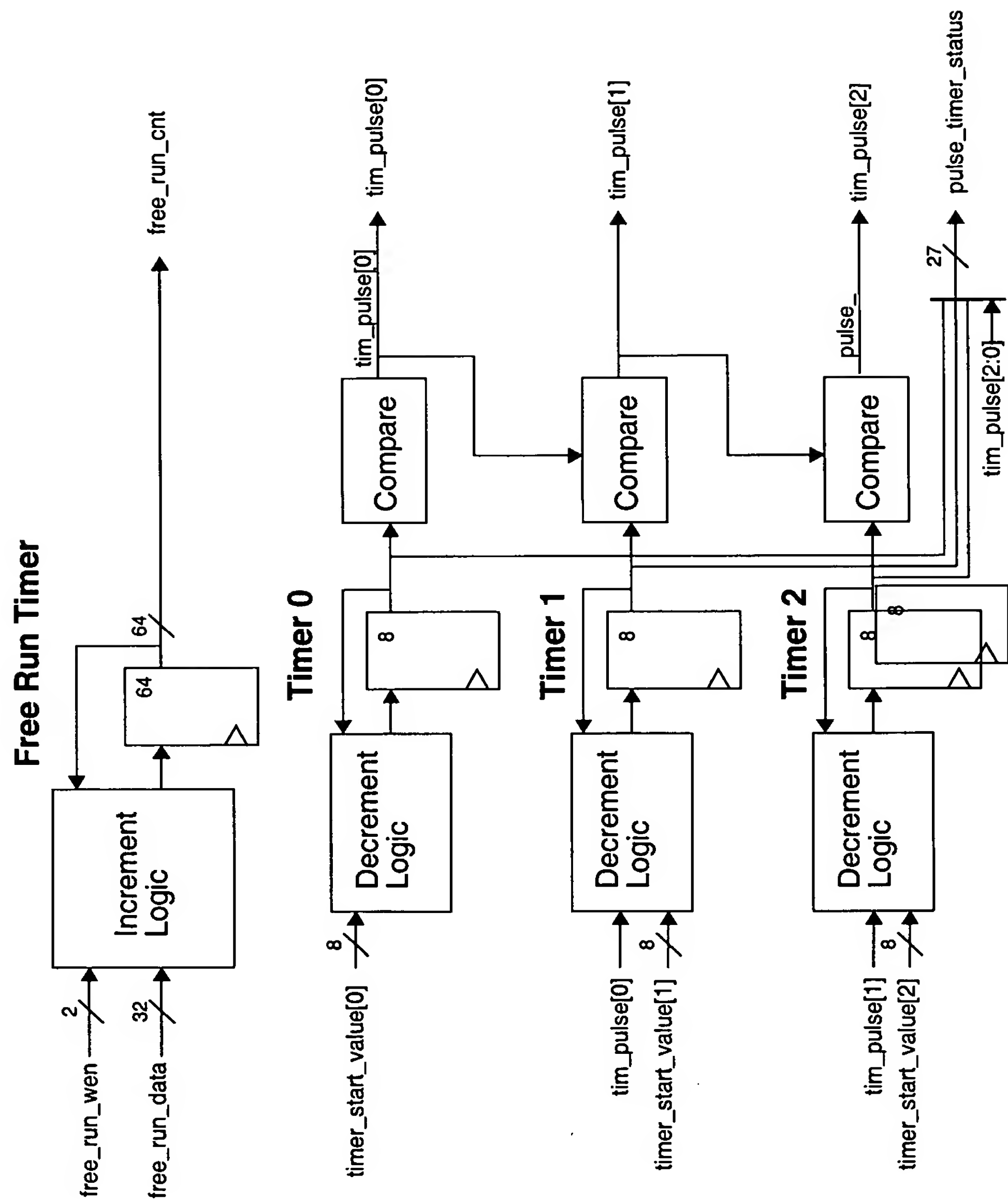


FIG. 66

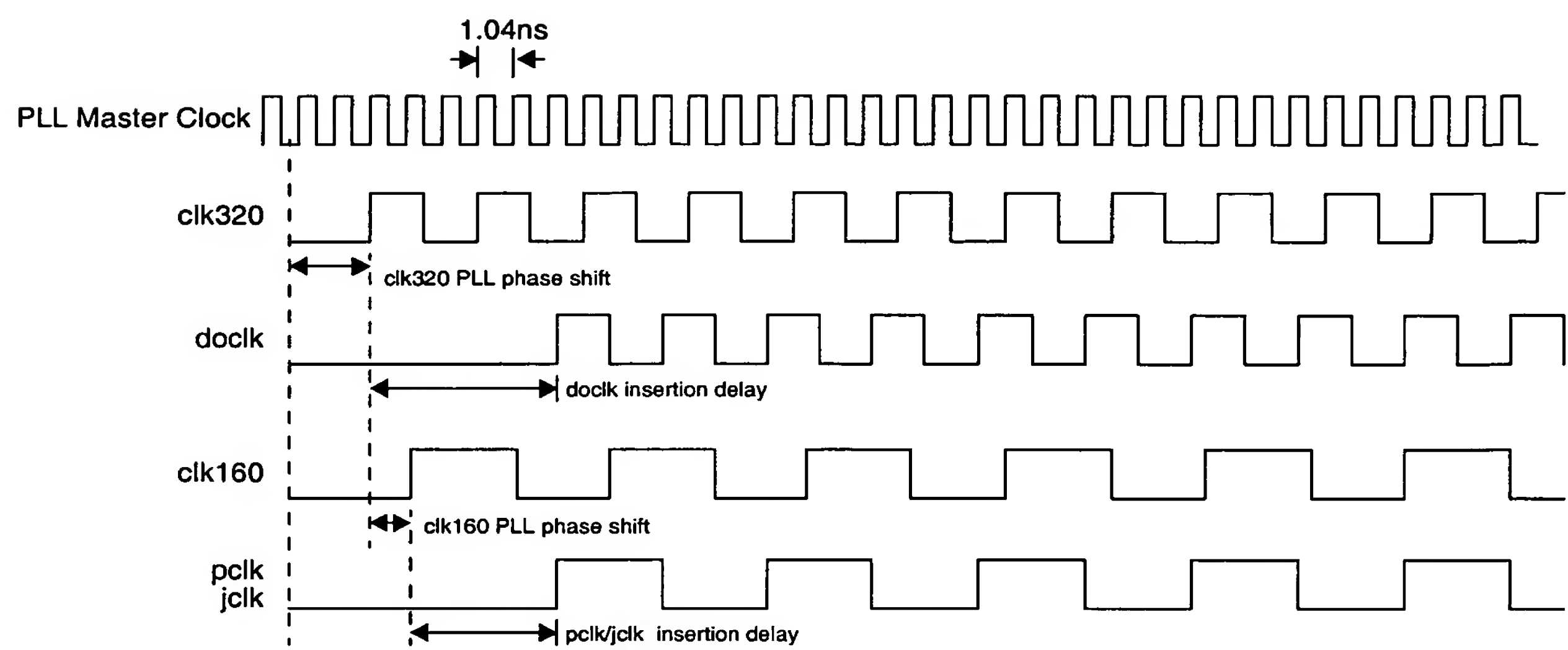


FIG. 67

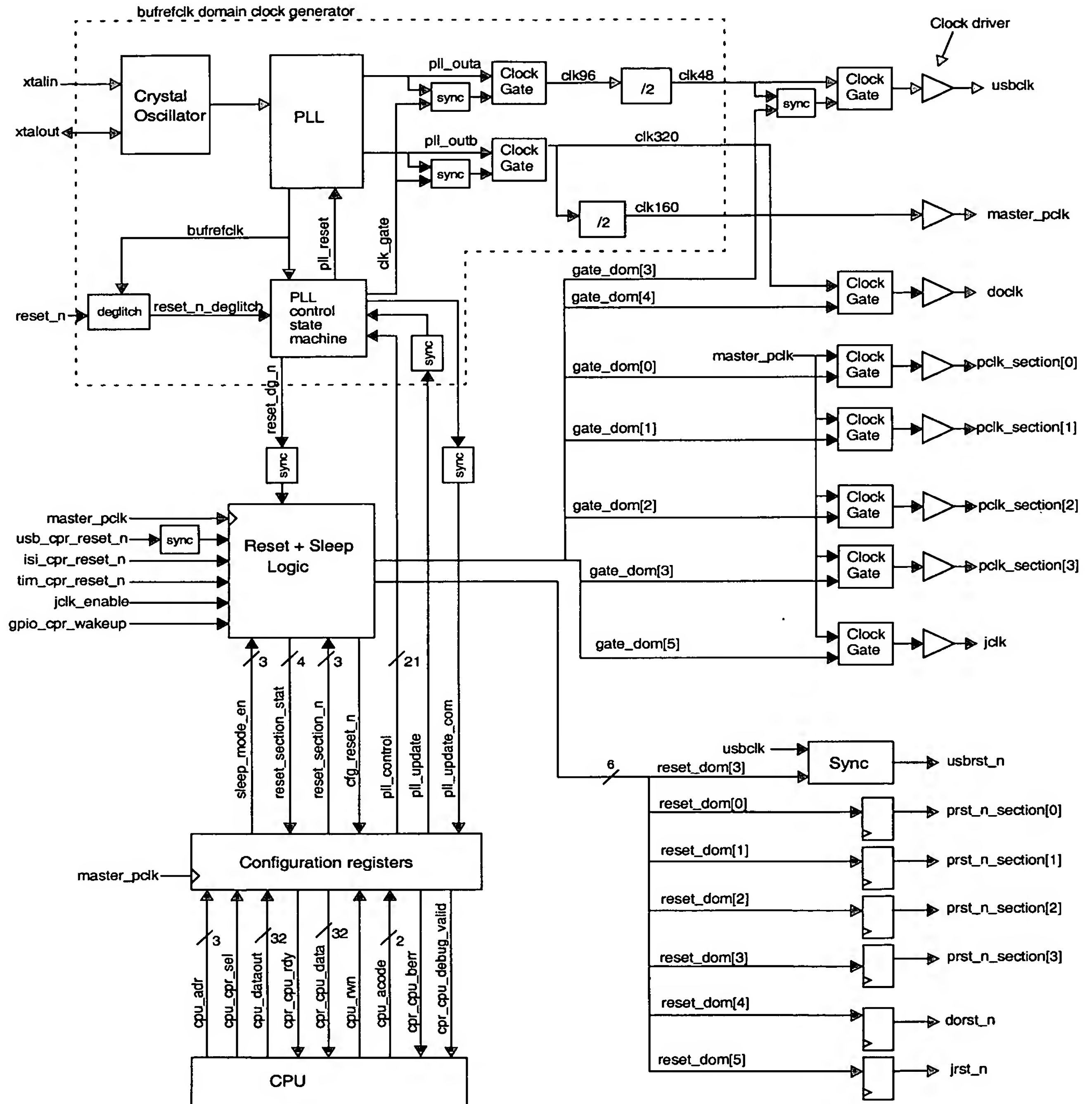
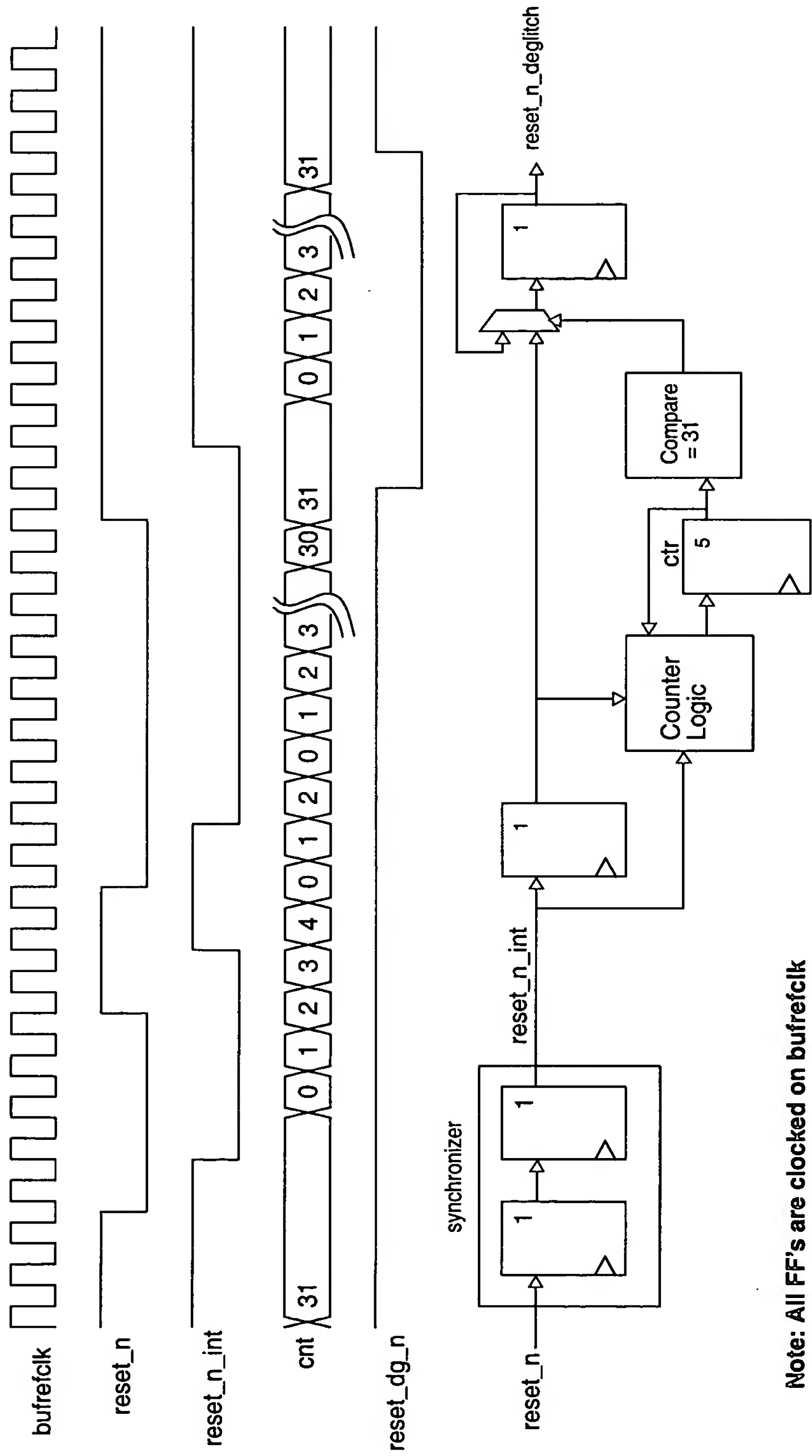


FIG. 68



Note: All FF's are clocked on bufrefclk

FIG. 69

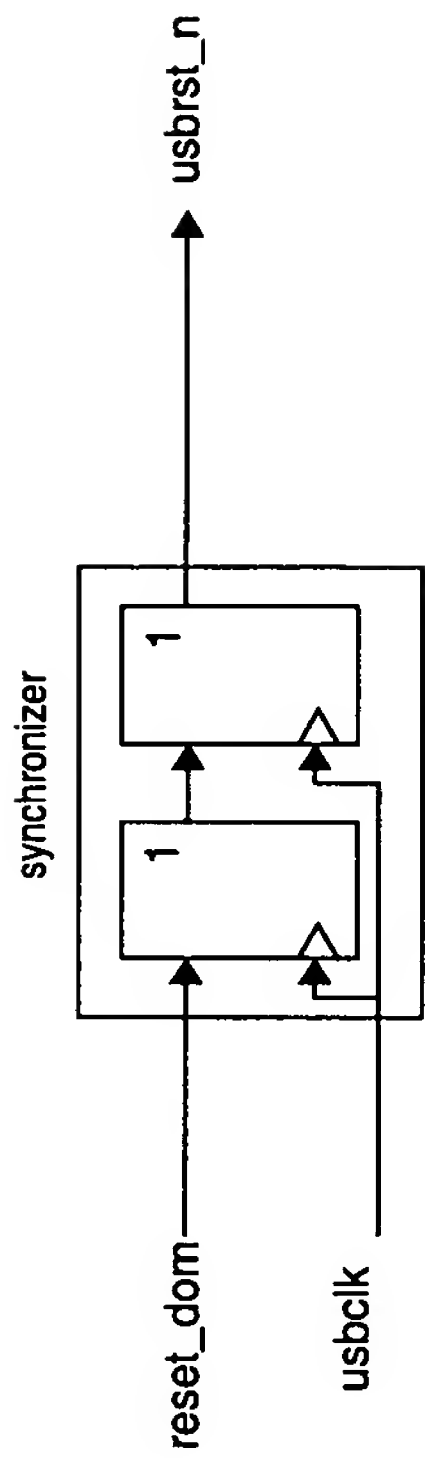
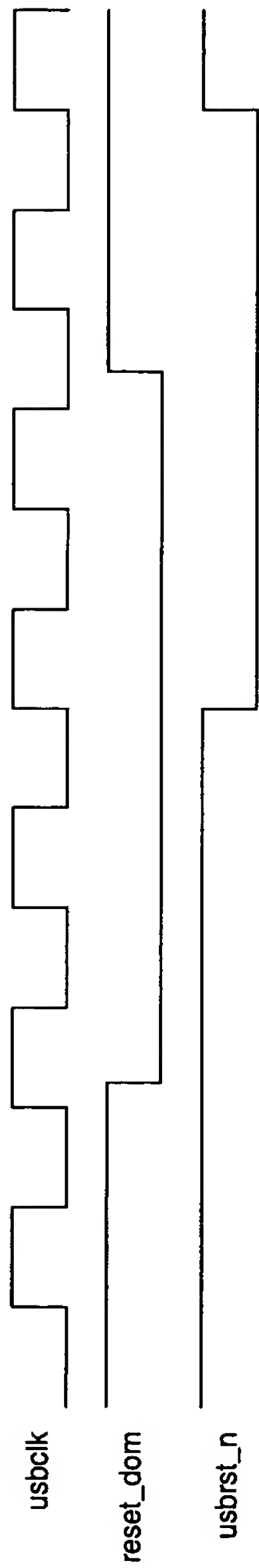


FIG. 70

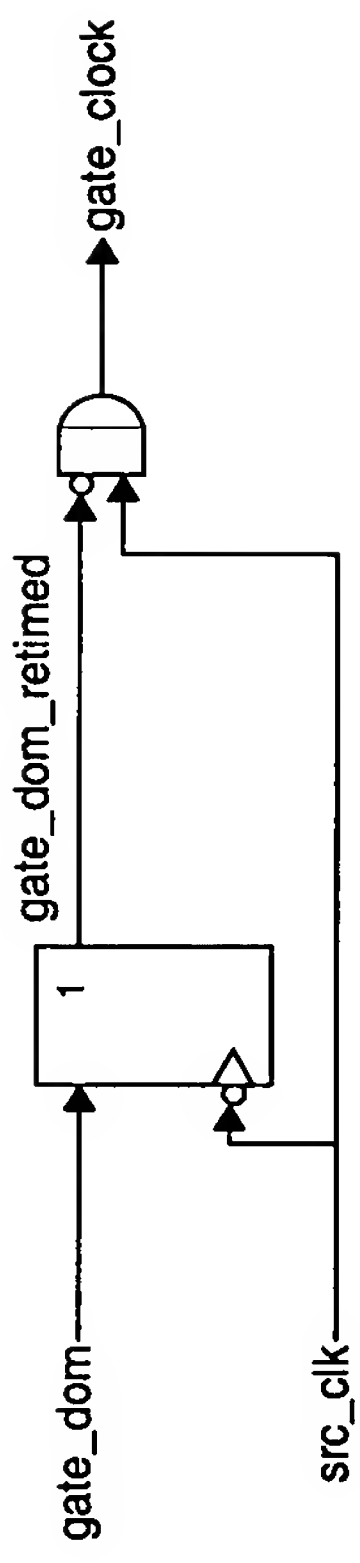
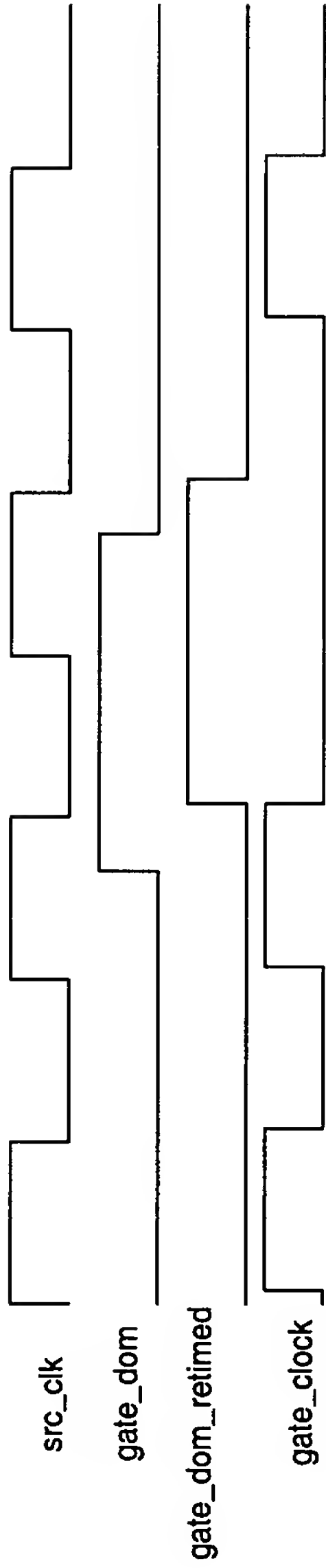
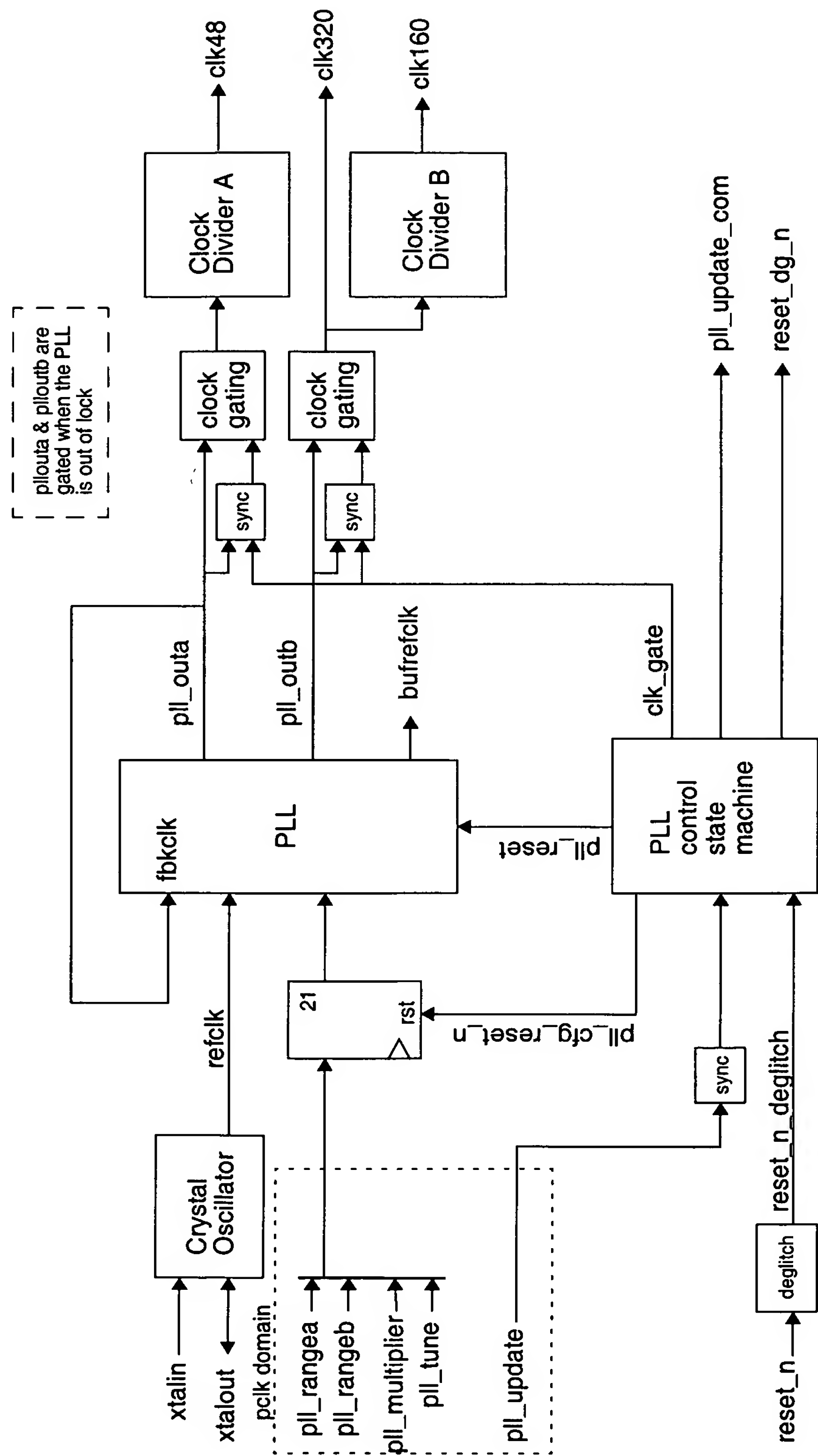


FIG. 71



Note: All logic clocked on bufrefclk unless otherwise indicated

FIG. 72

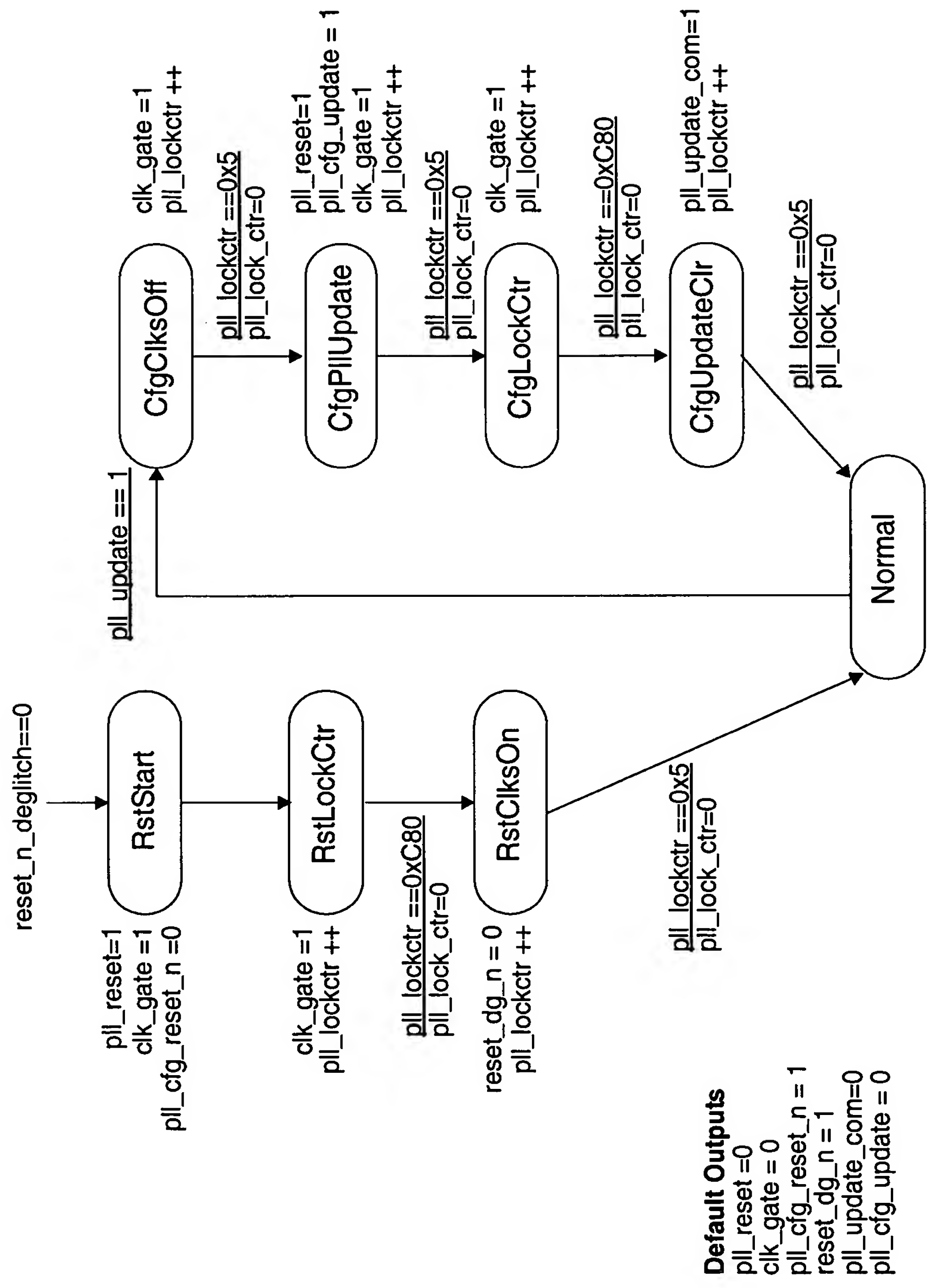


FIG. 73

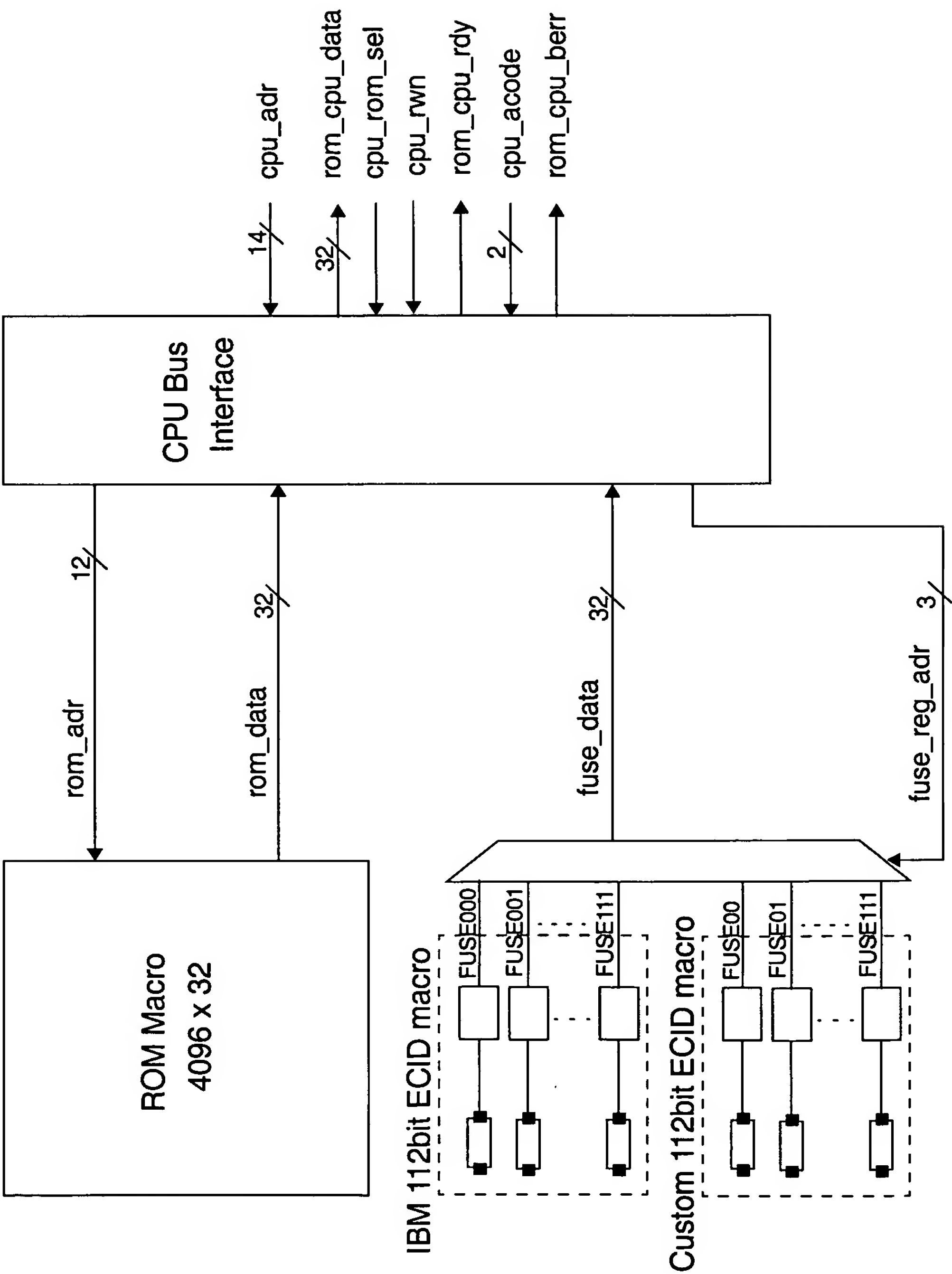


FIG. 74

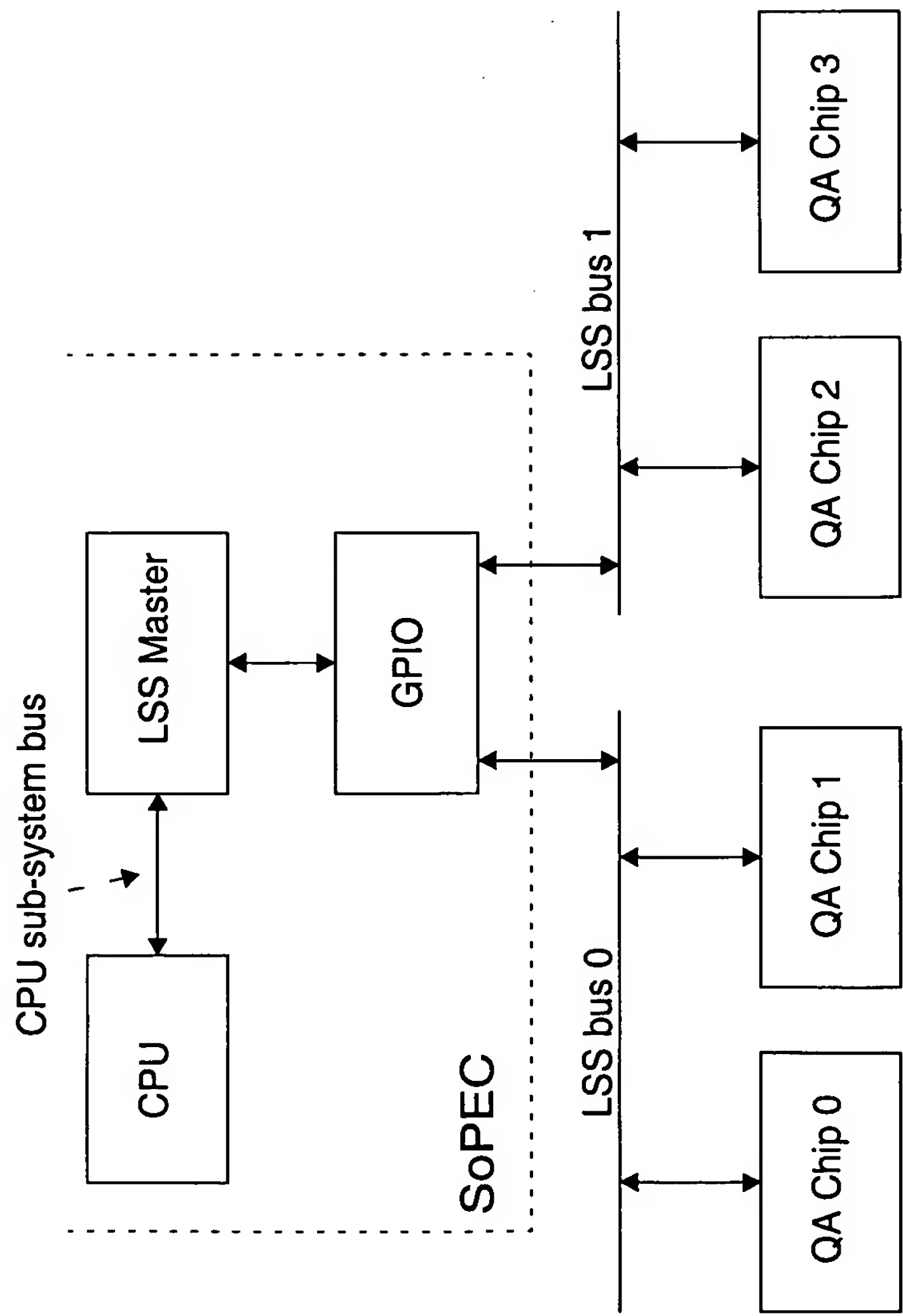


FIG. 75

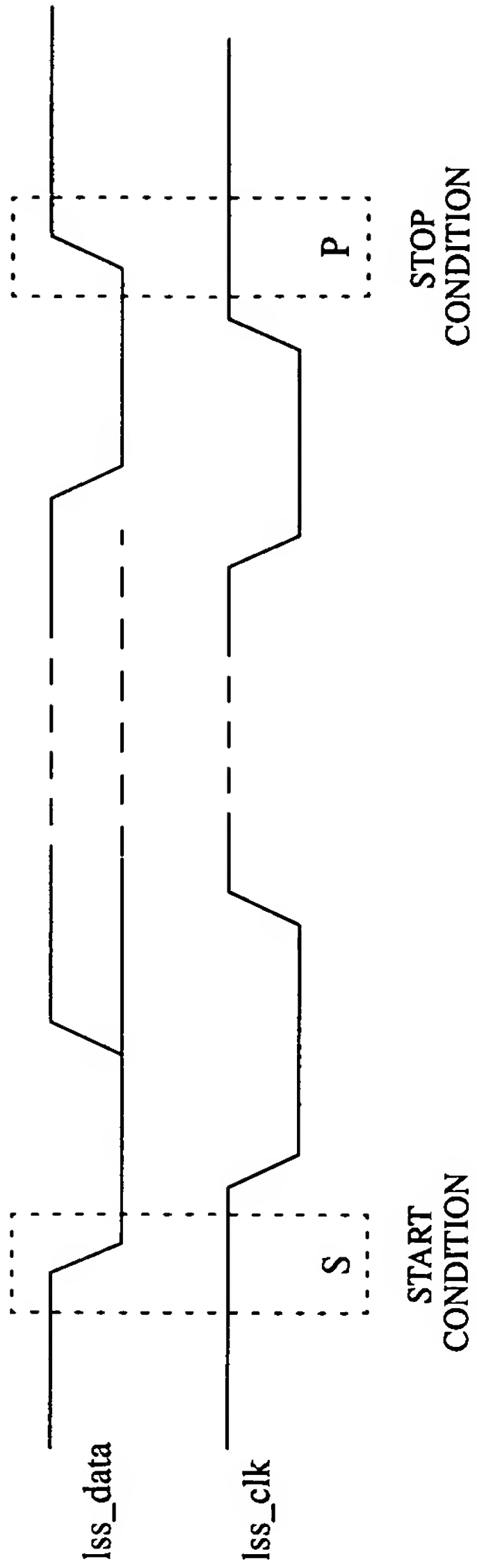


FIG. 76

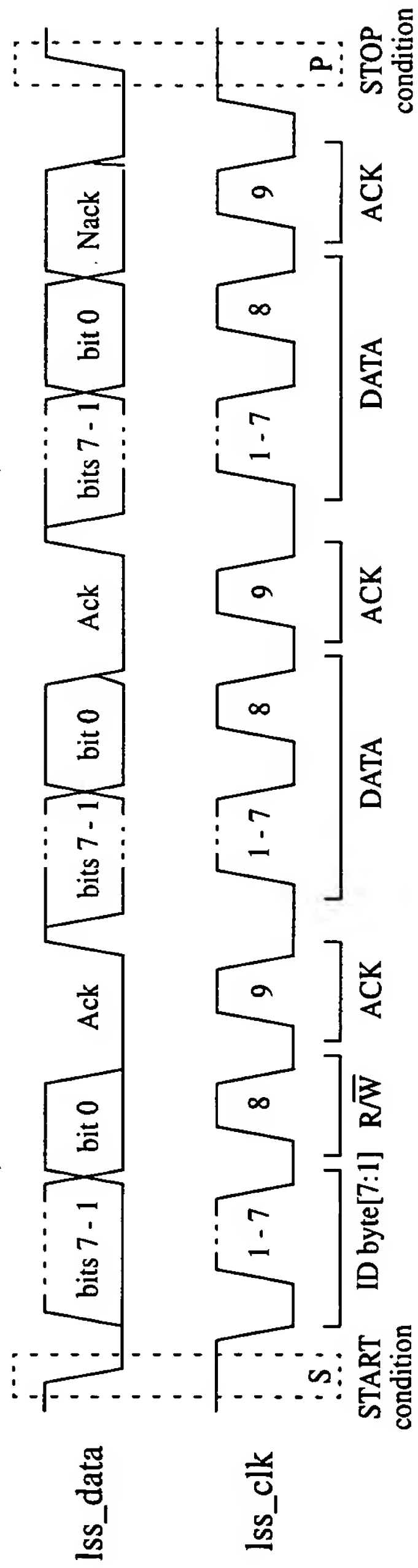


FIG. 77

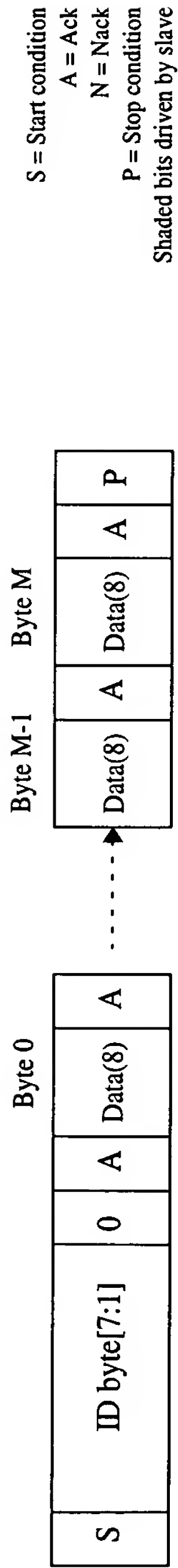


FIG. 78

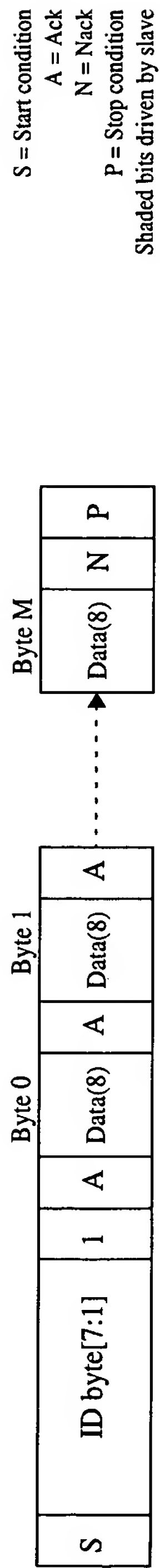


FIG. 79

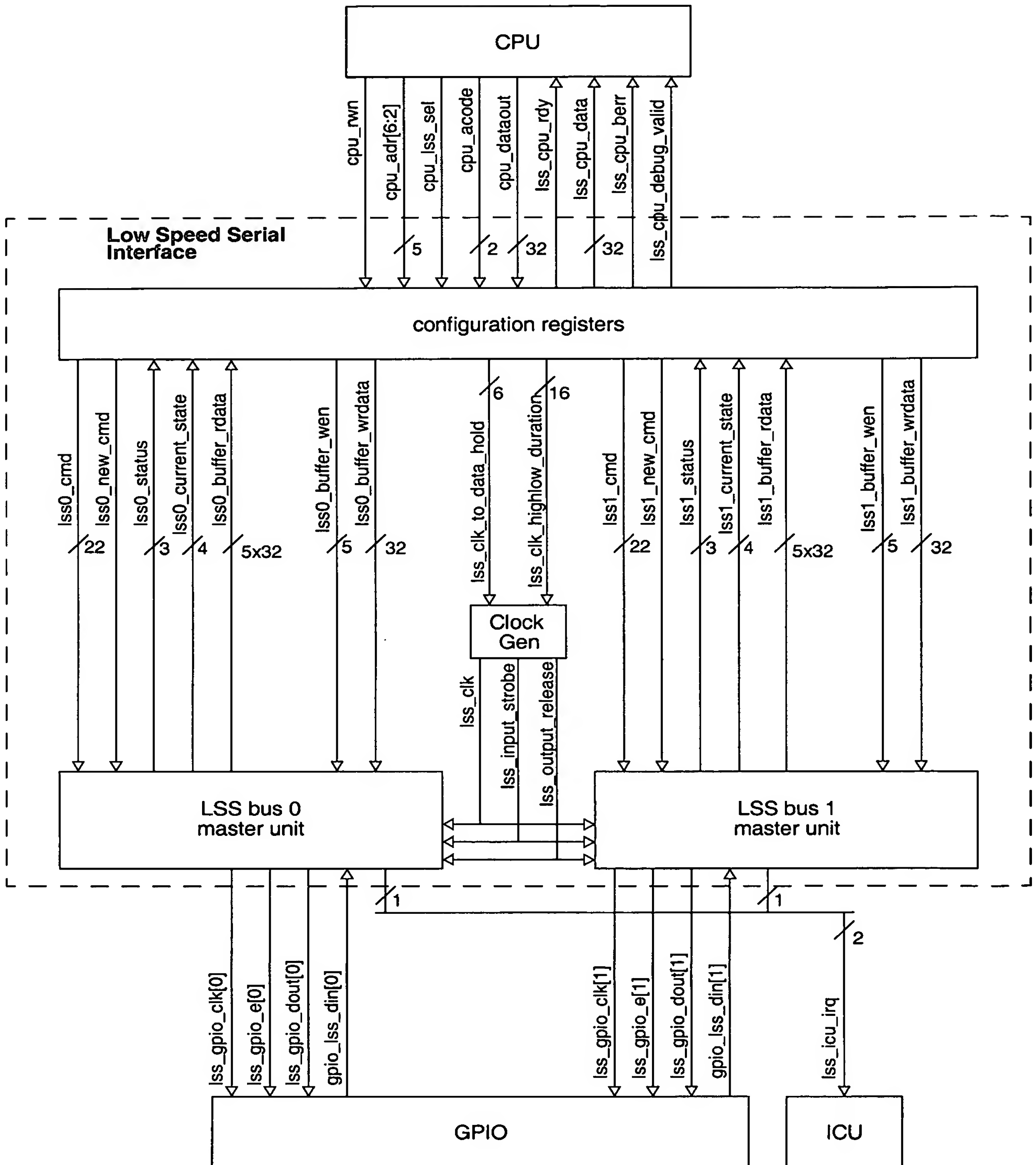


FIG. 80

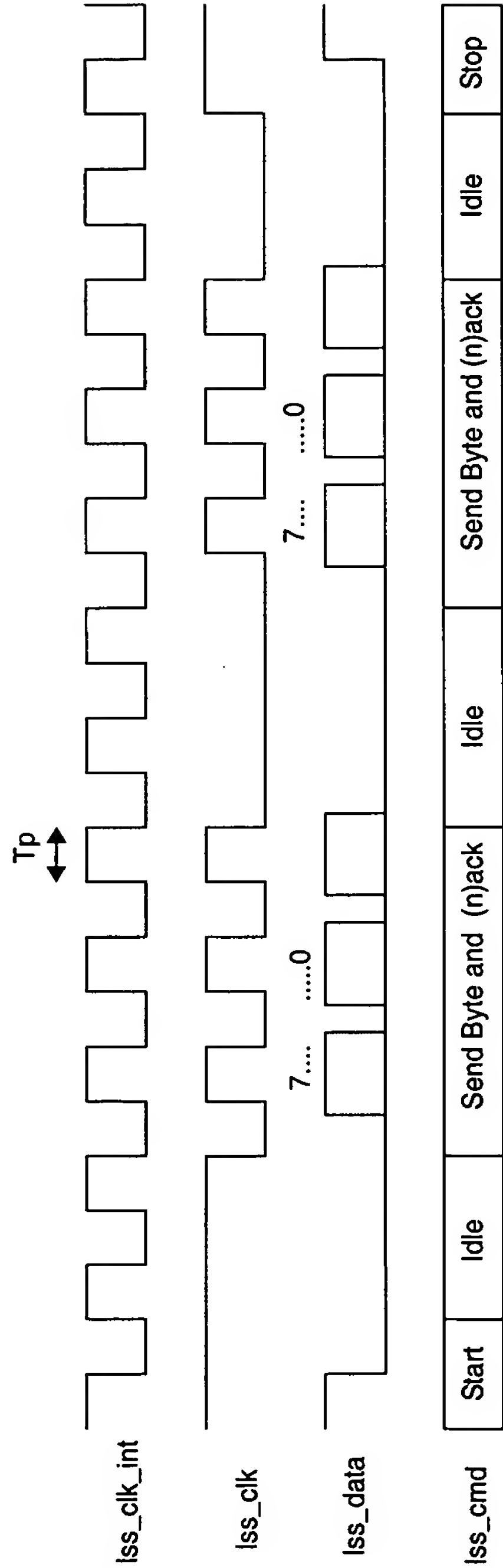


FIG. 81

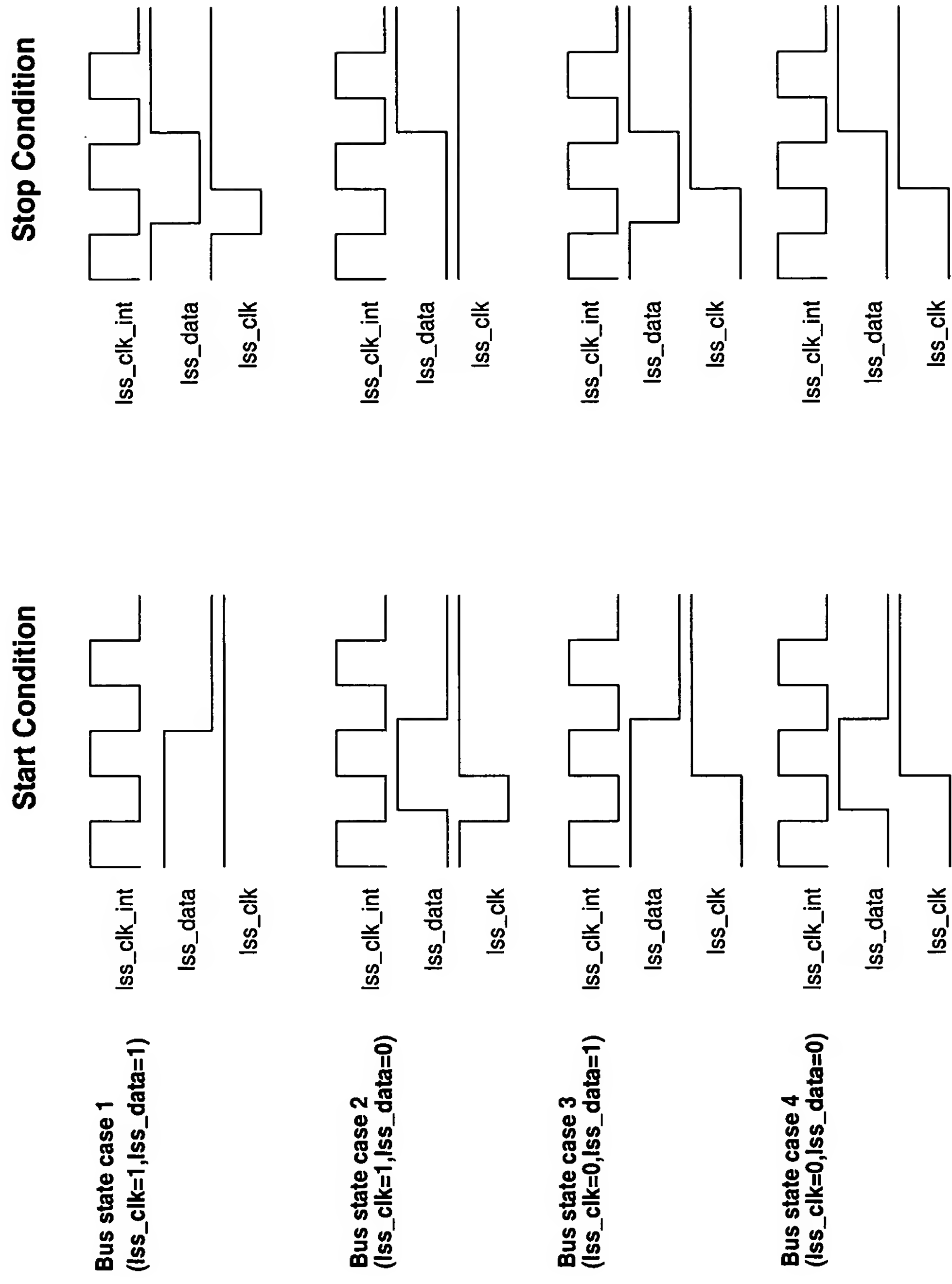


FIG. 82

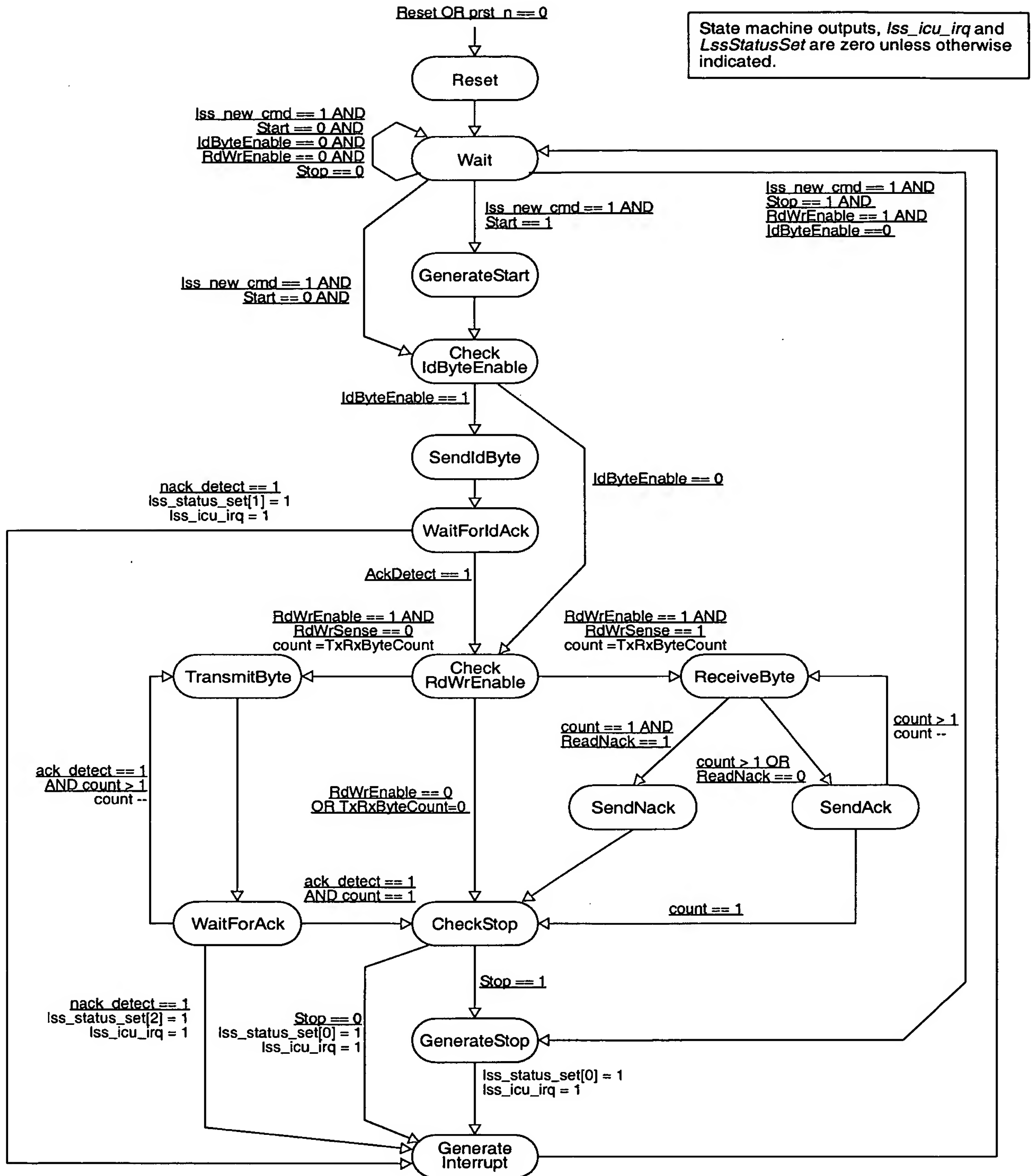


FIG. 83

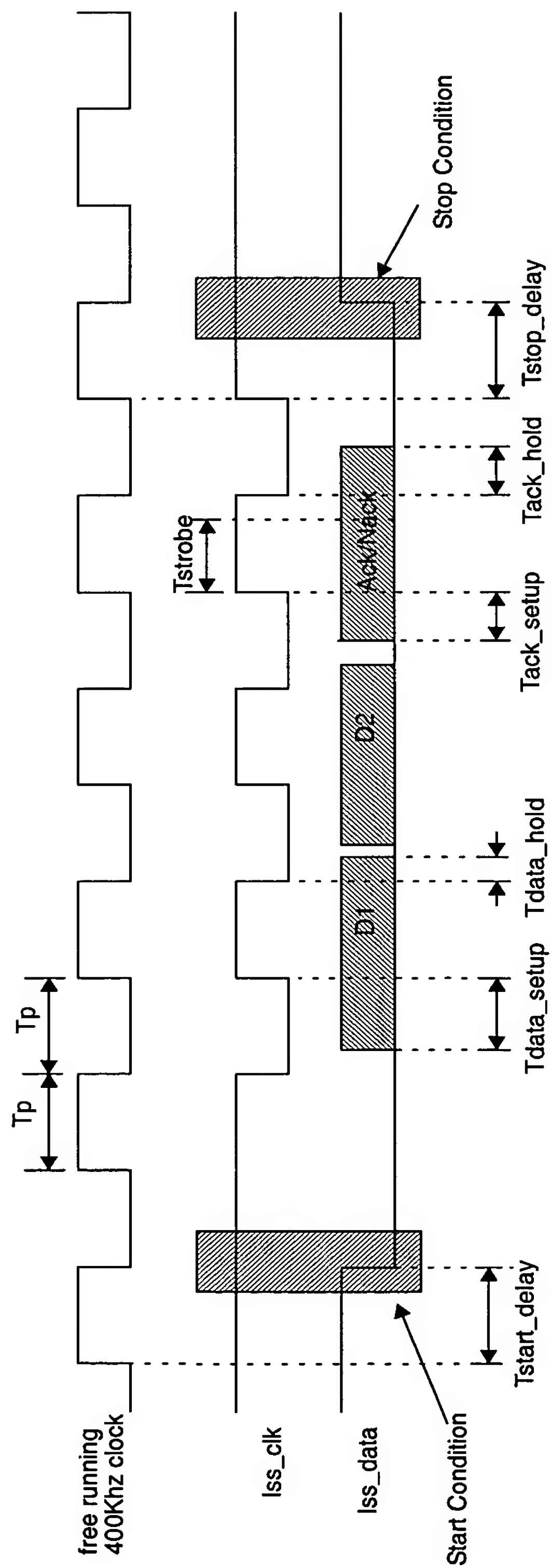


FIG. 84

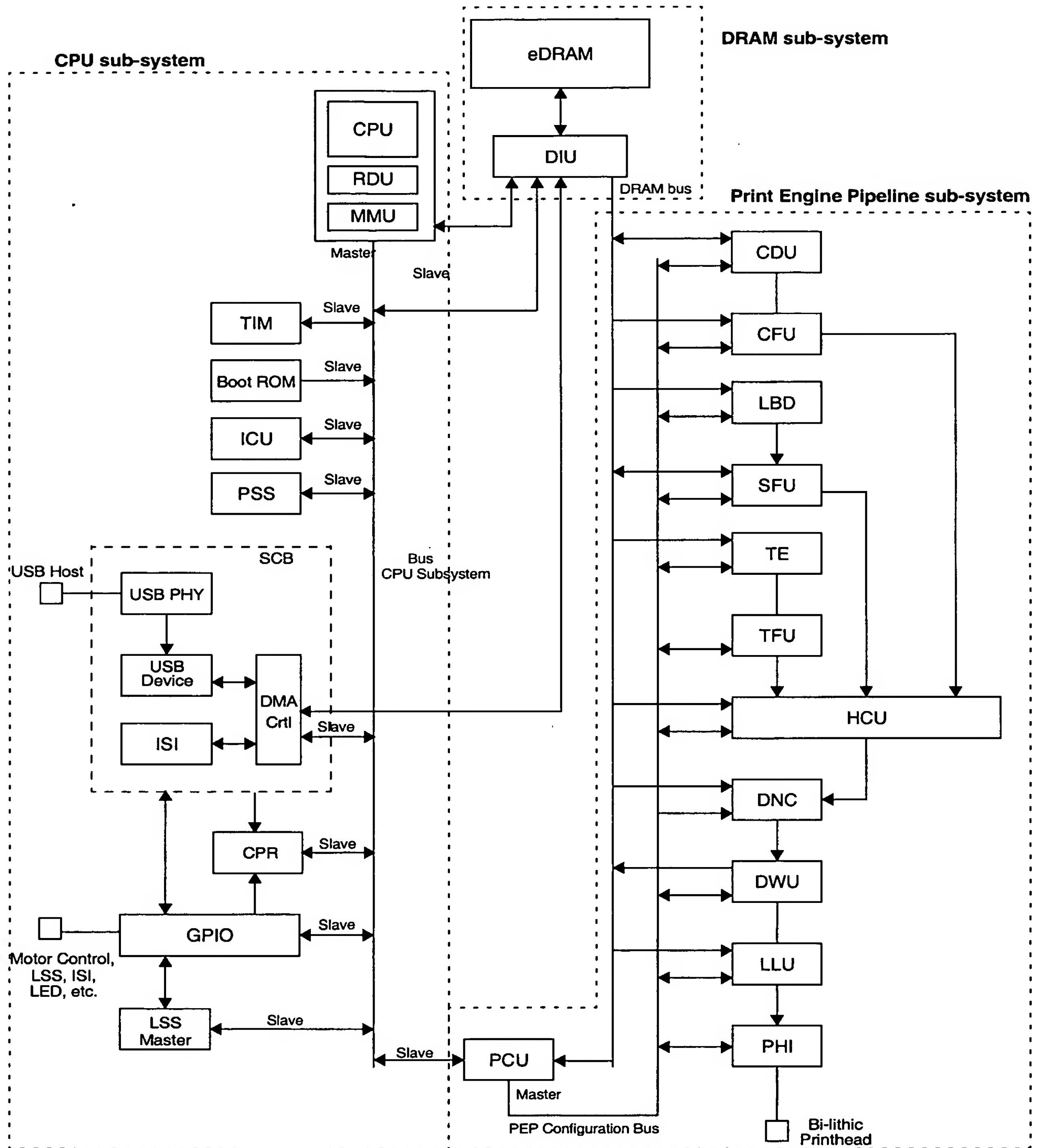


FIG. 85

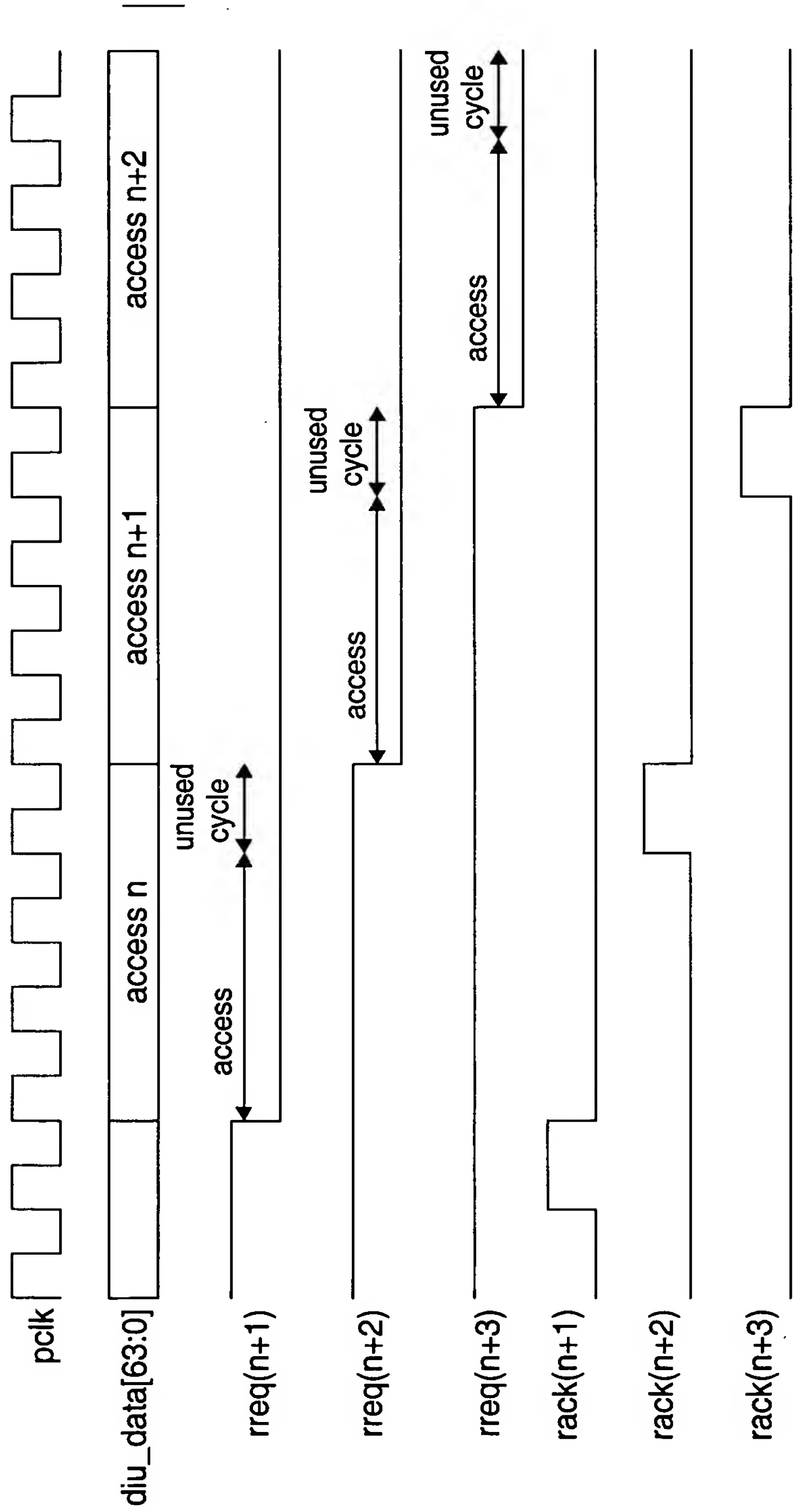


FIG. 86

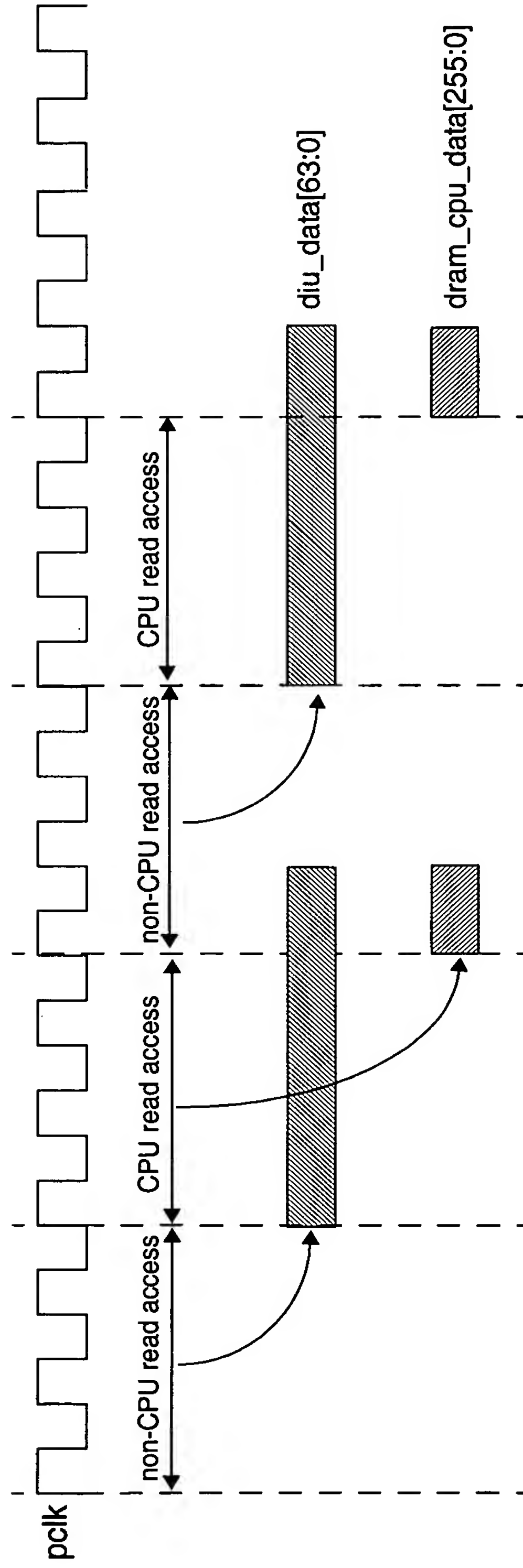


FIG. 87

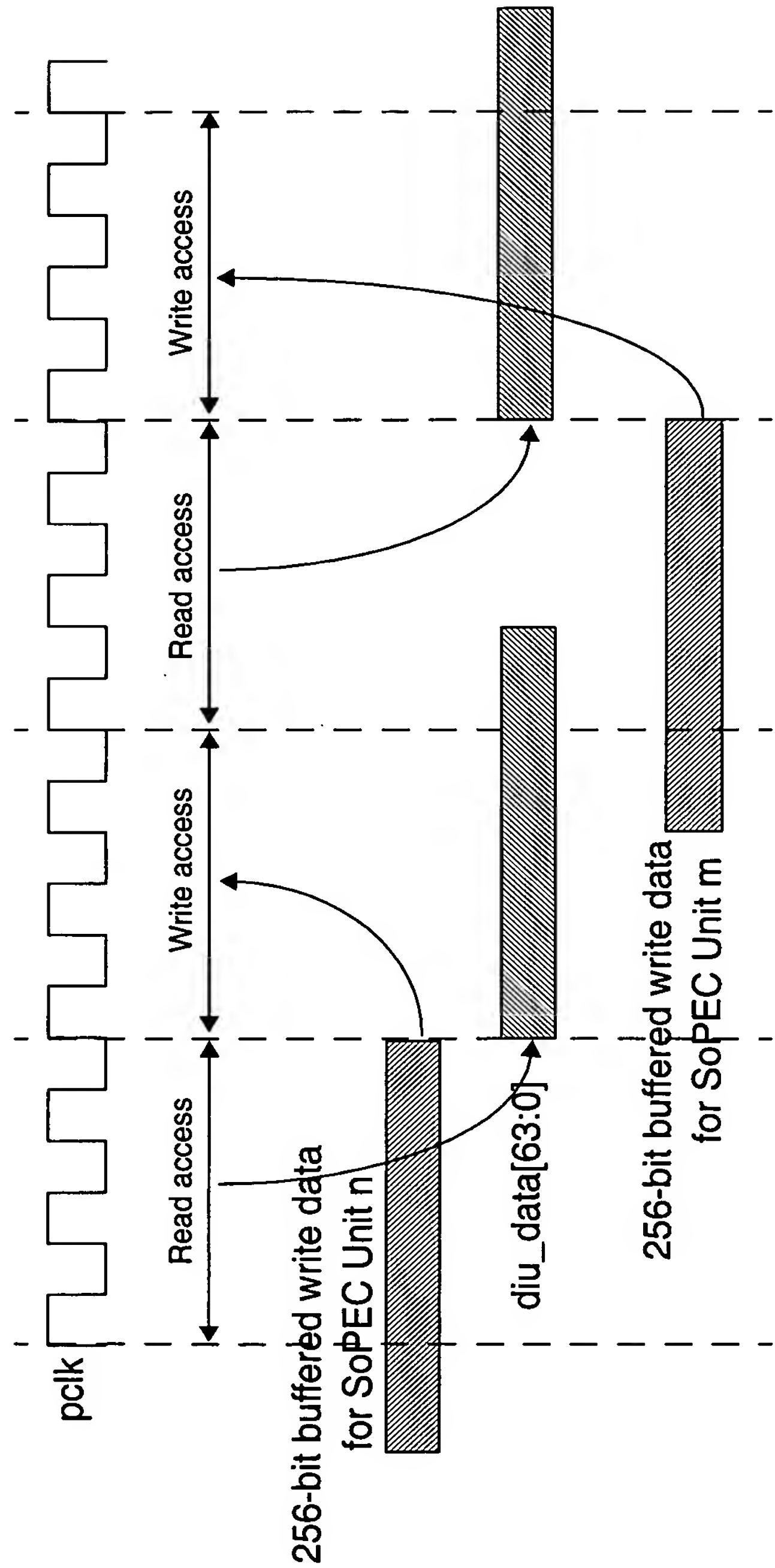
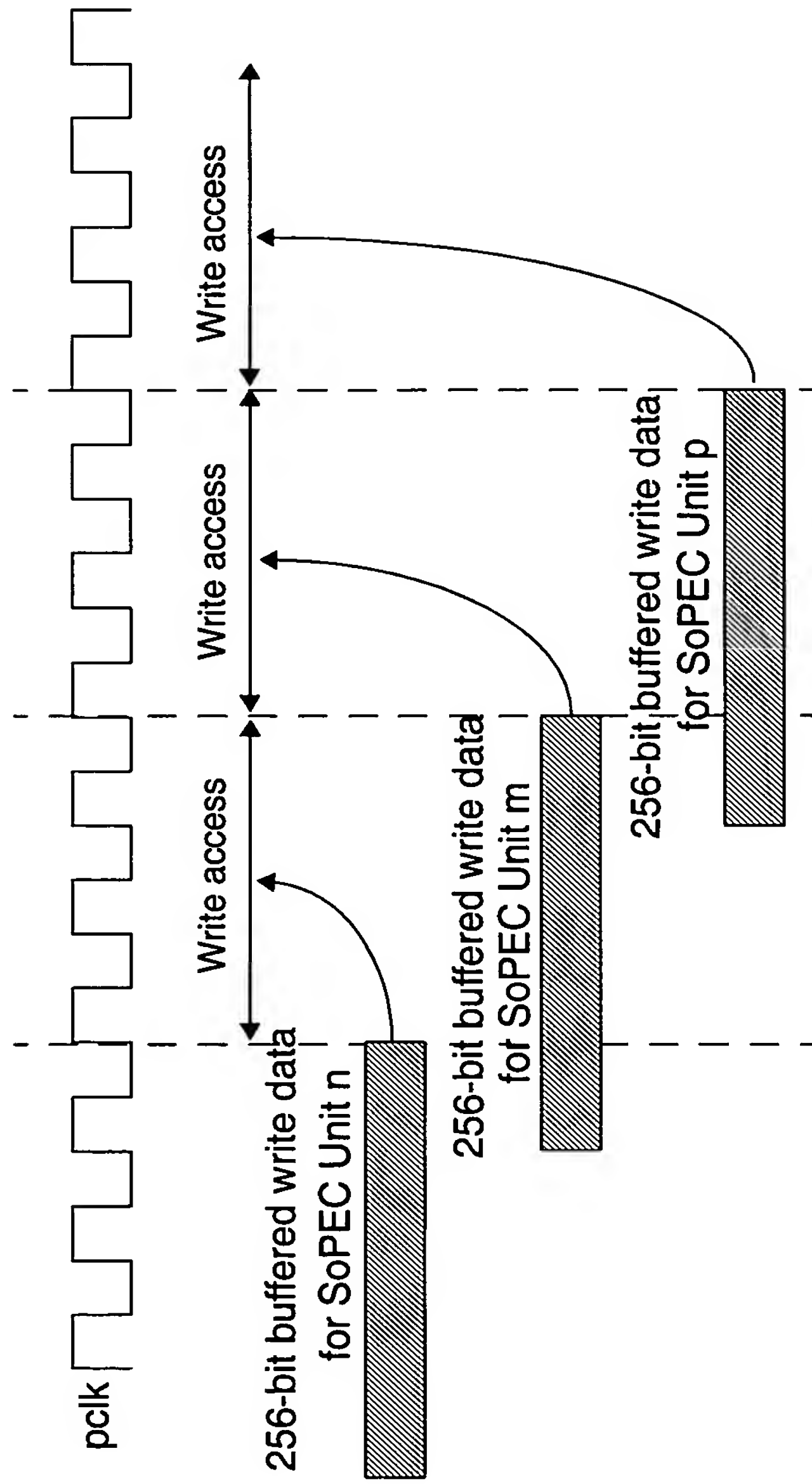


FIG. 88

*FIG. 89*

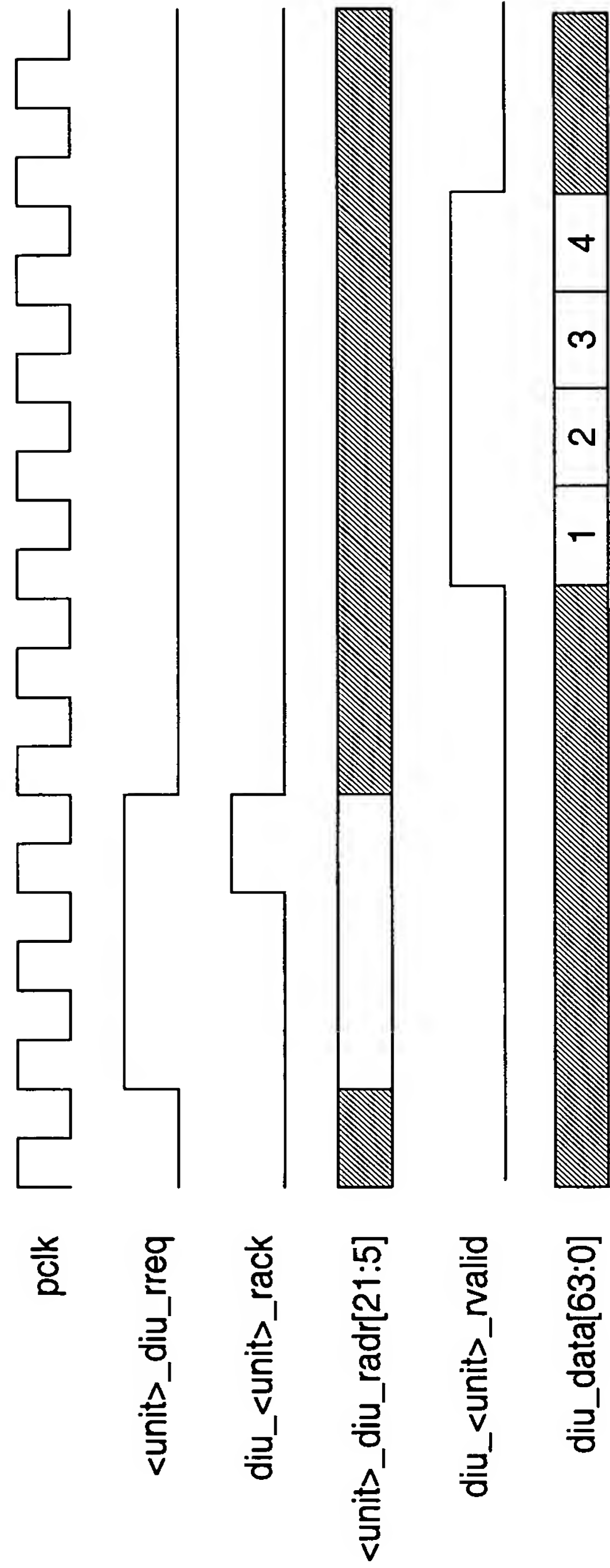


FIG. 90

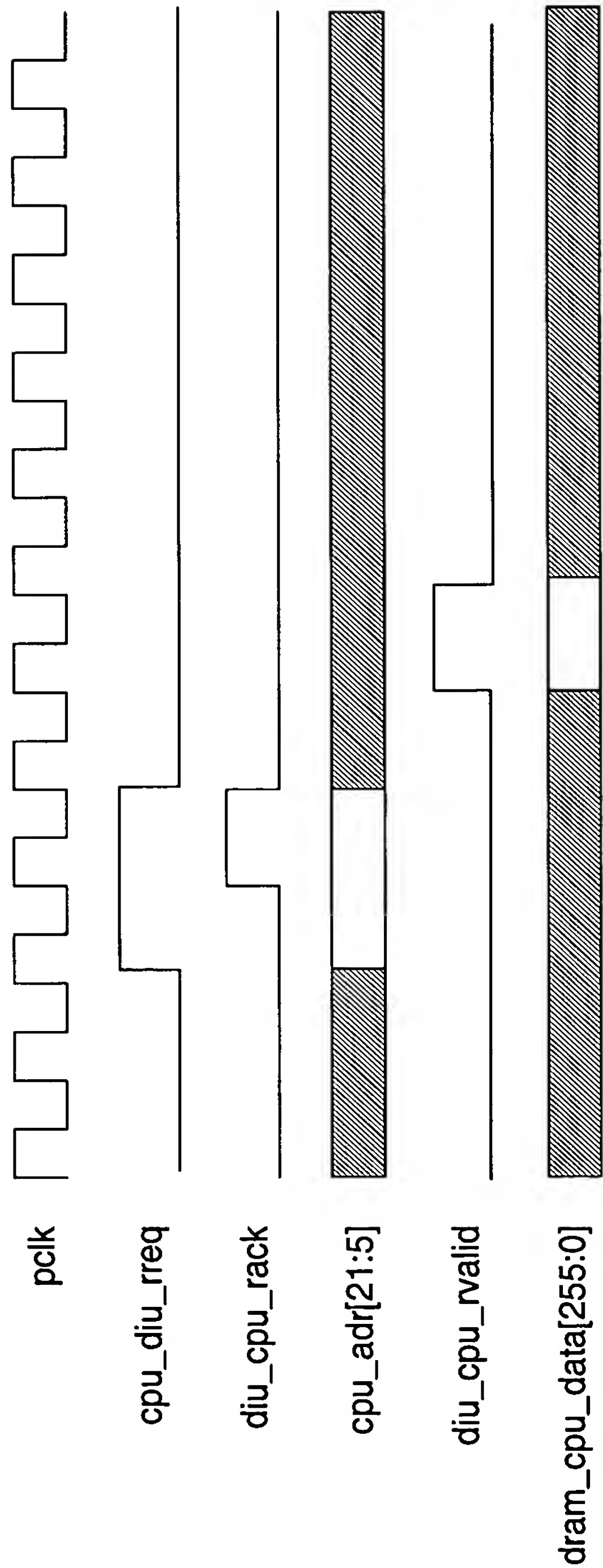


FIG. 91

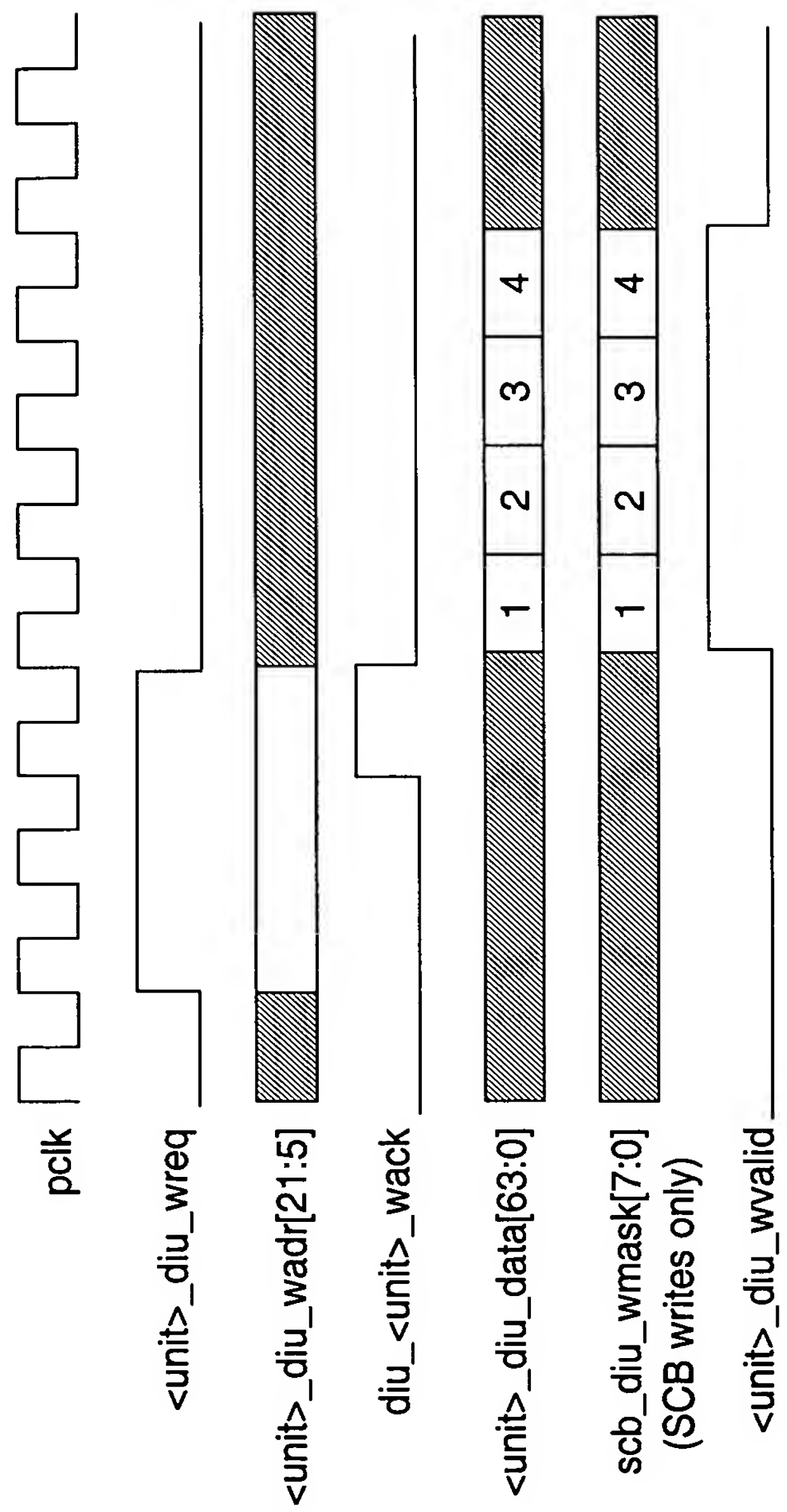


FIG. 92

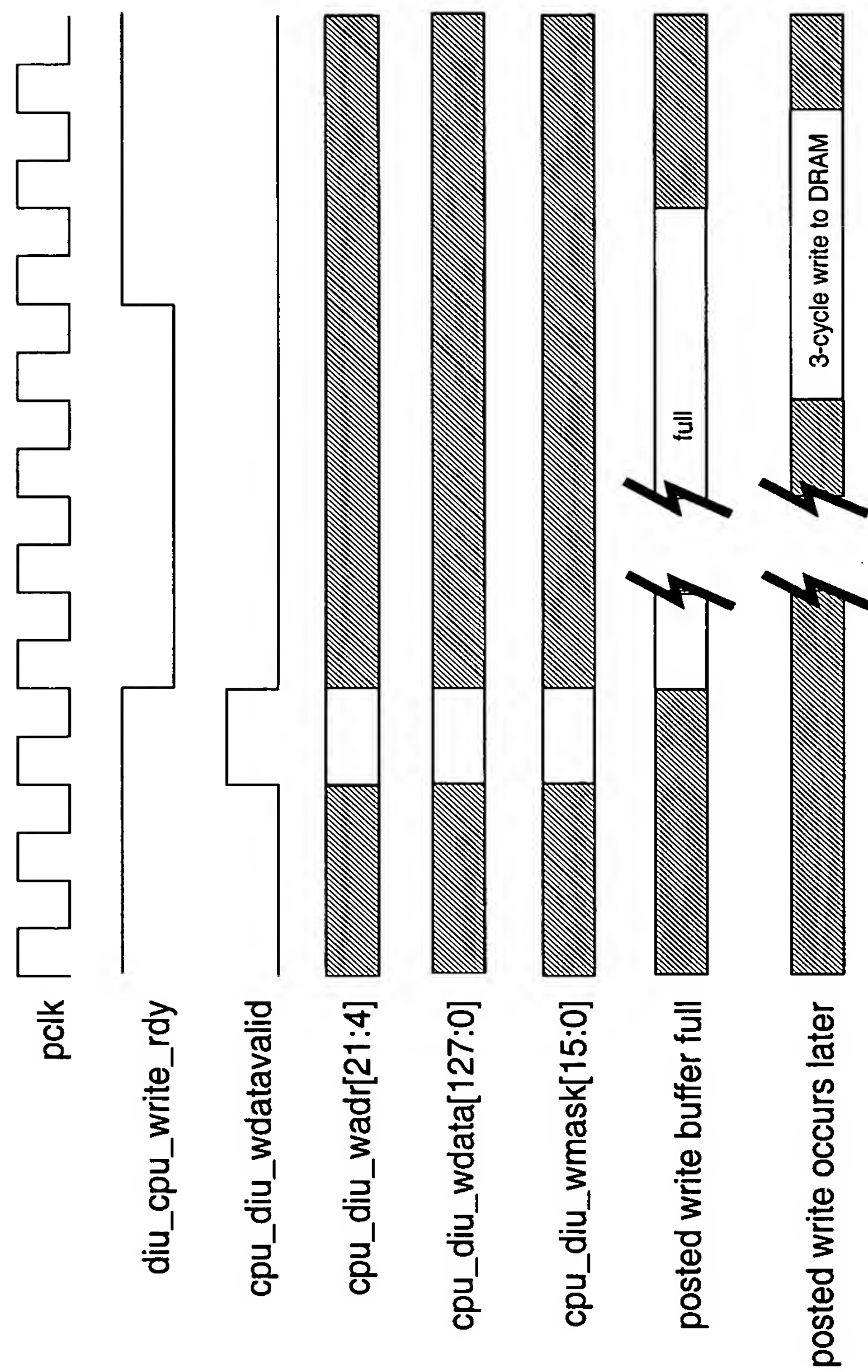


FIG. 93

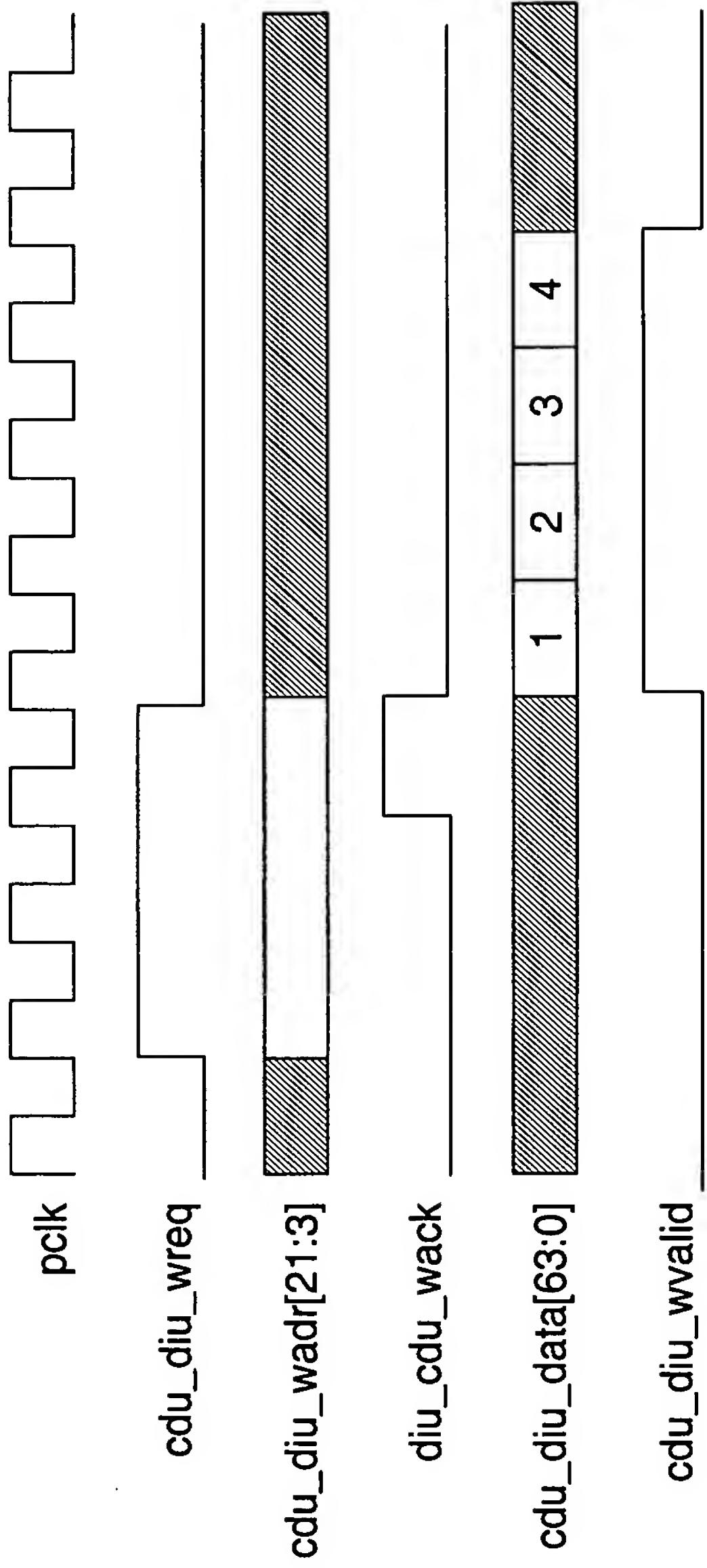
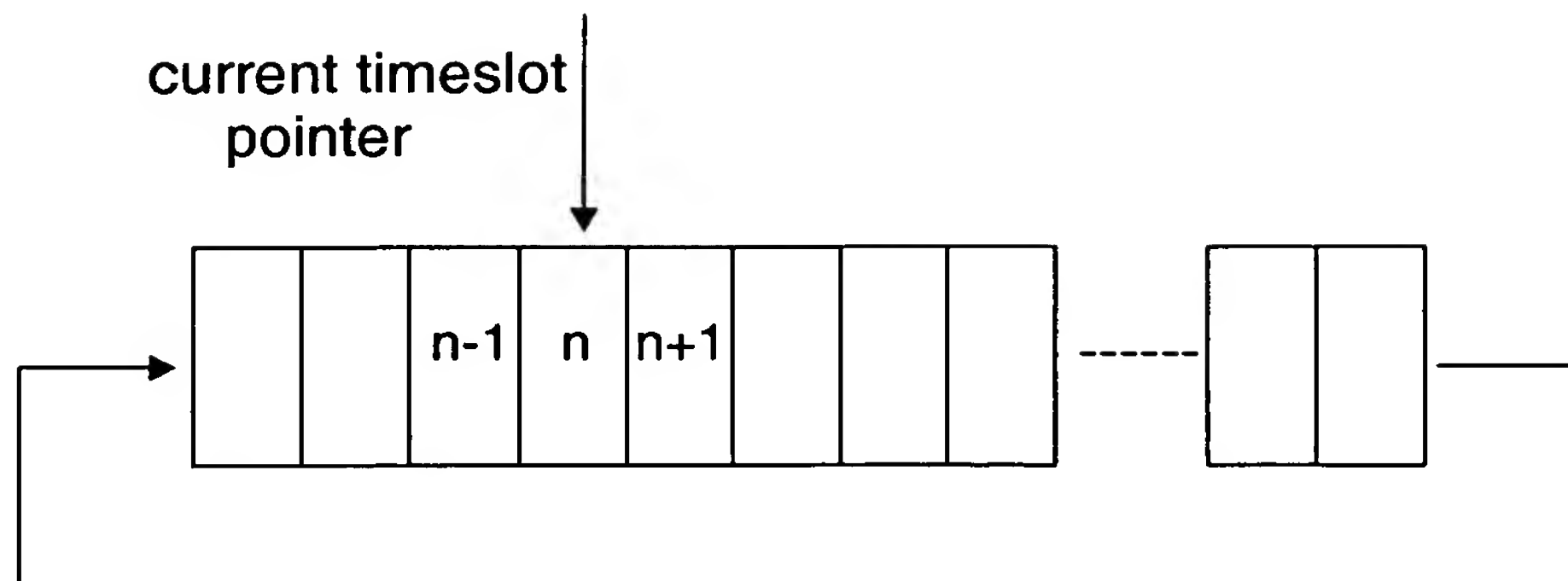
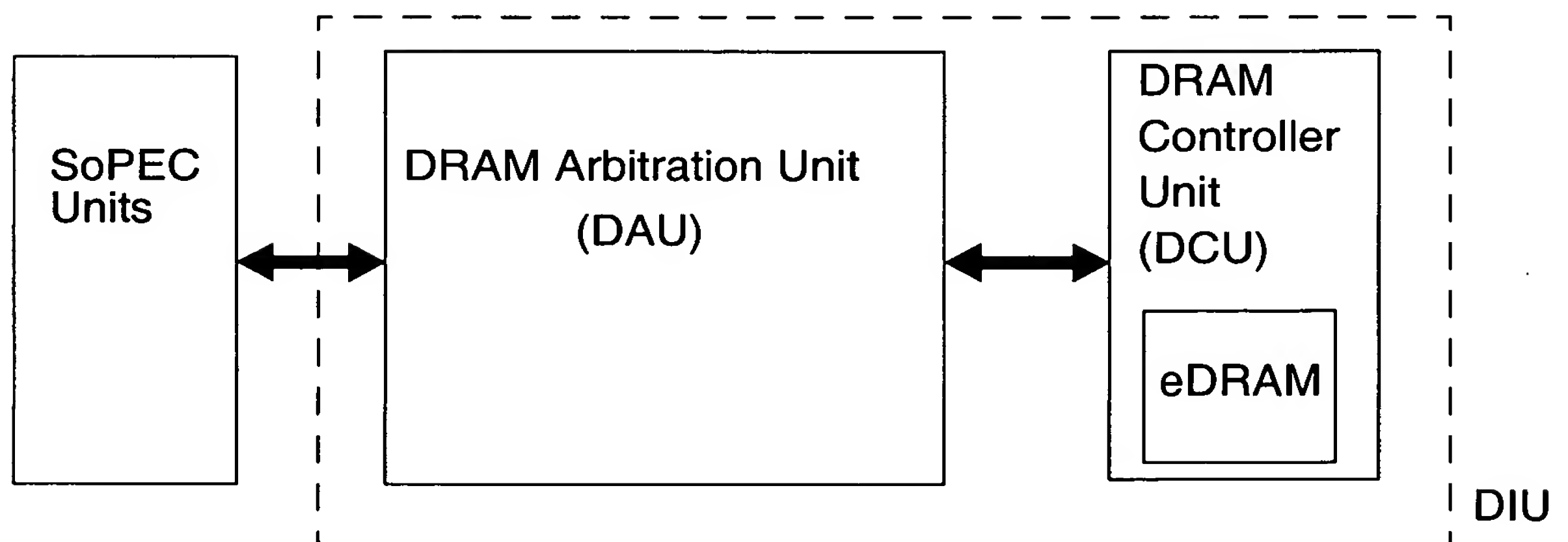


FIG. 94

*FIG. 95**FIG. 100*

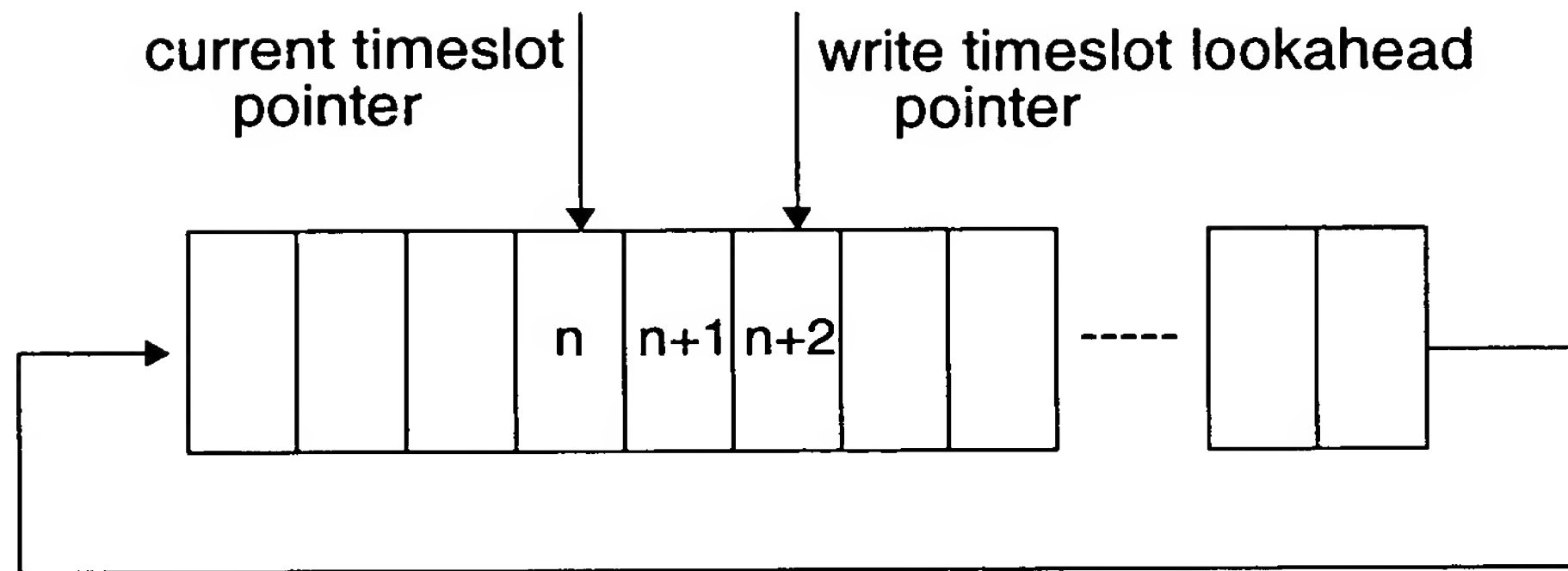


FIG. 96

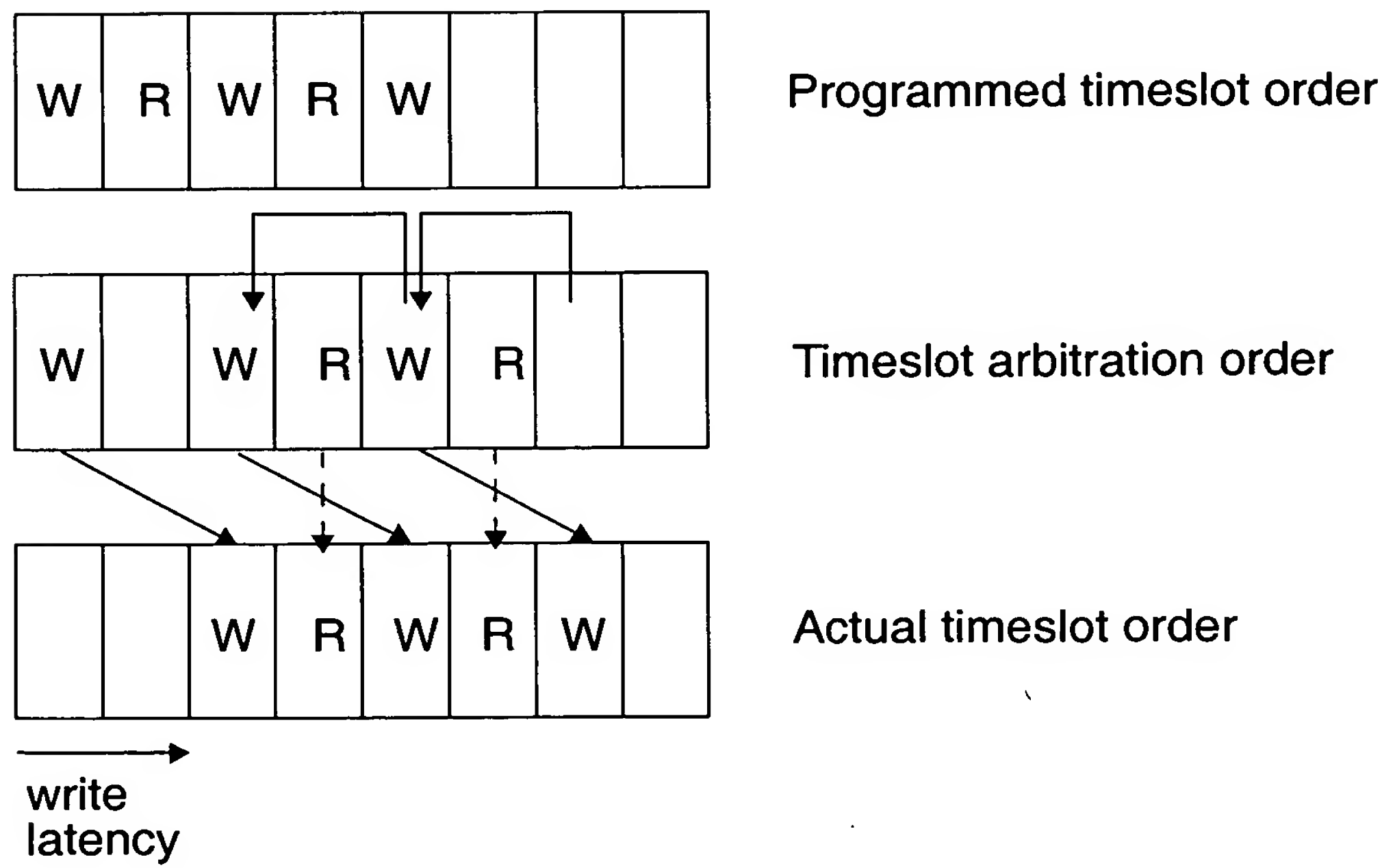


FIG. 97

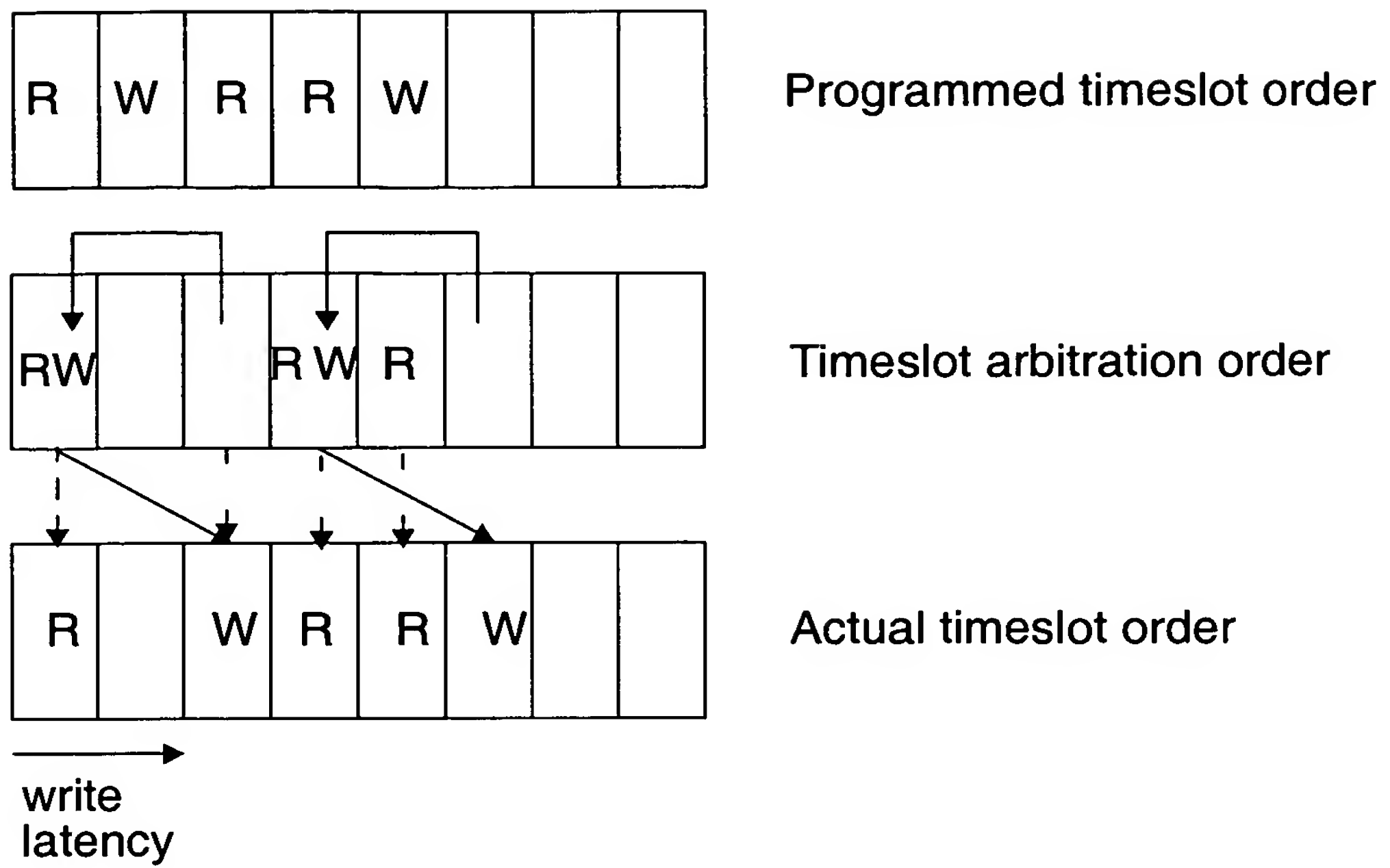


FIG. 98

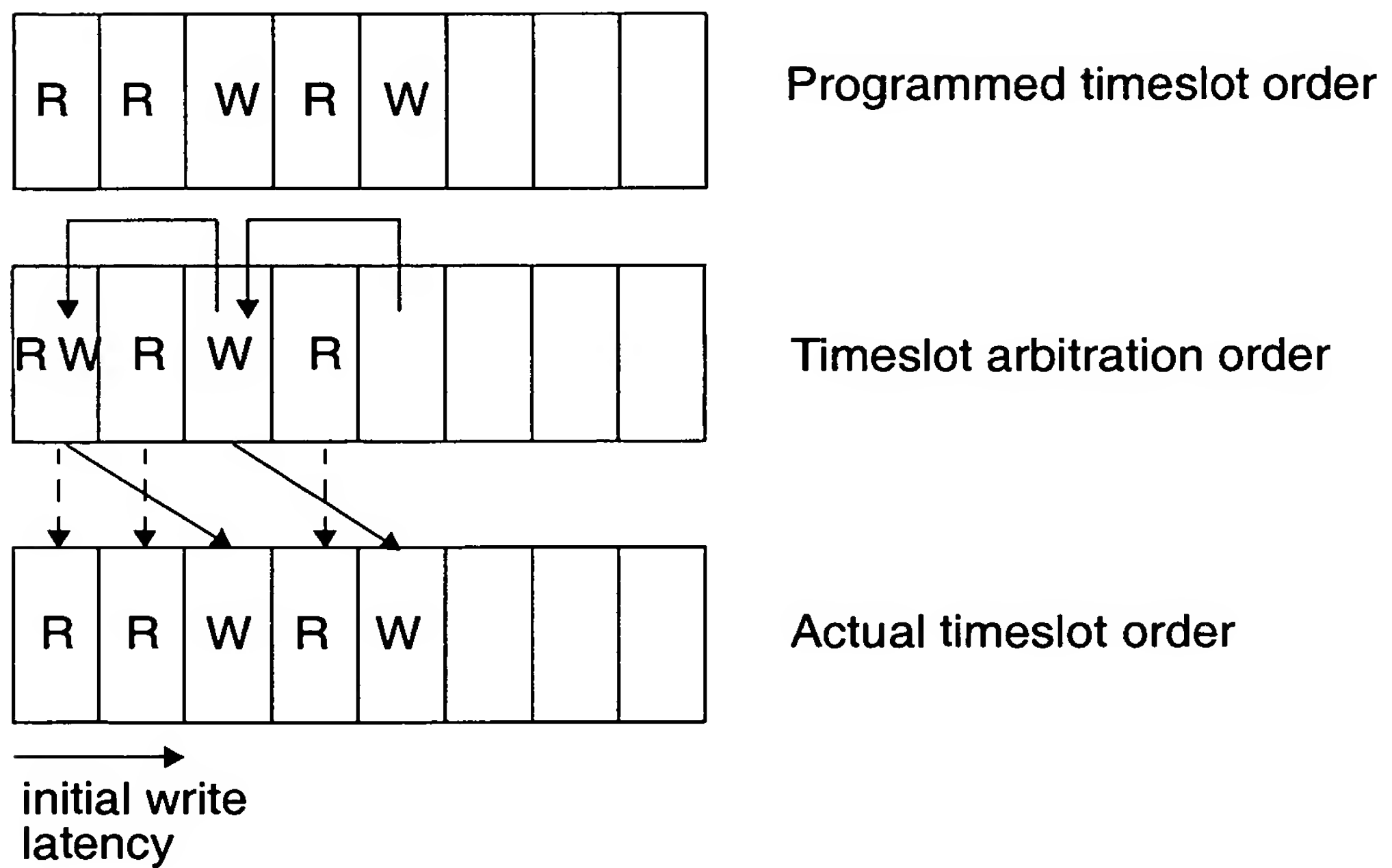


FIG. 99

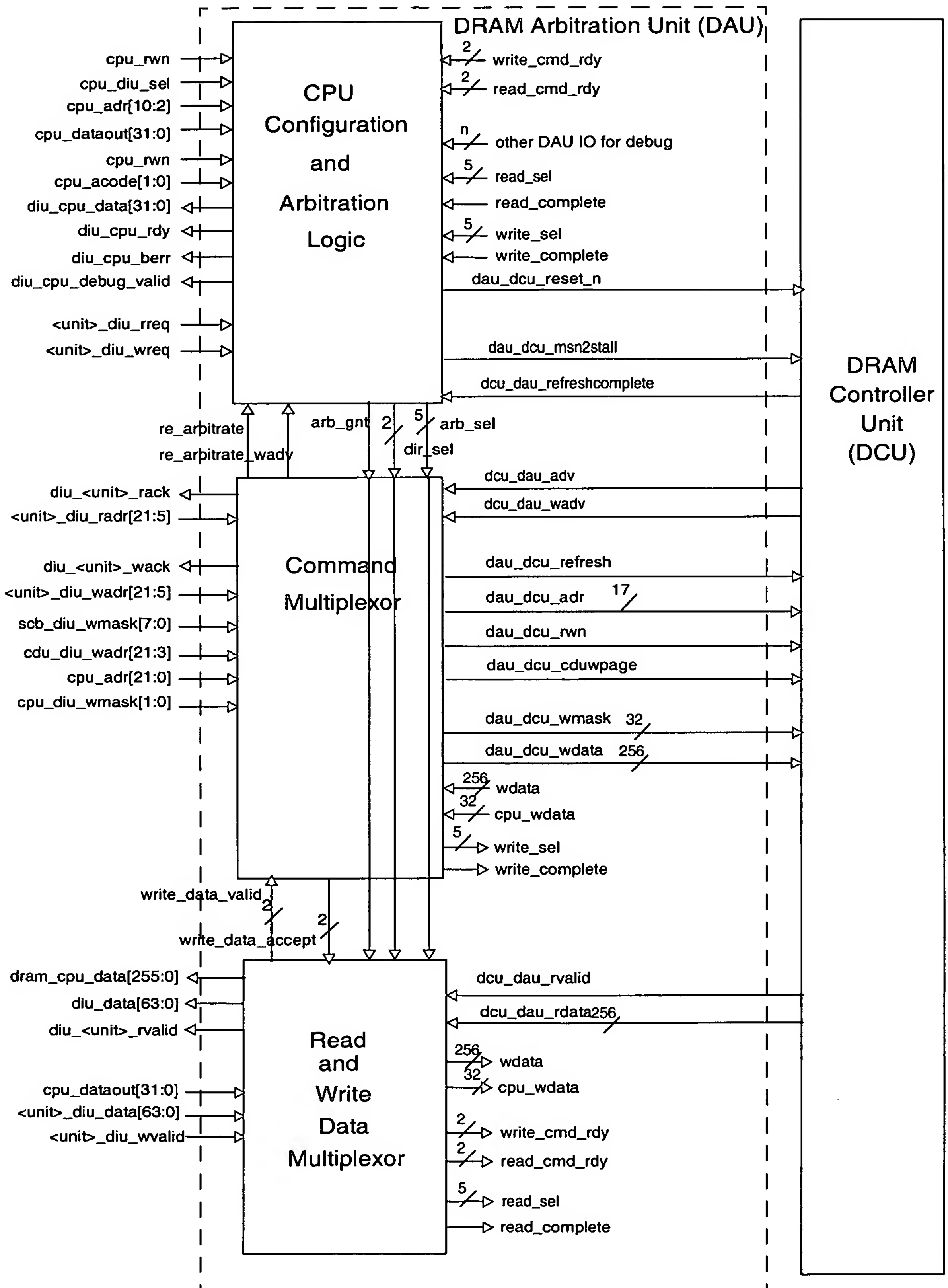


FIG. 101

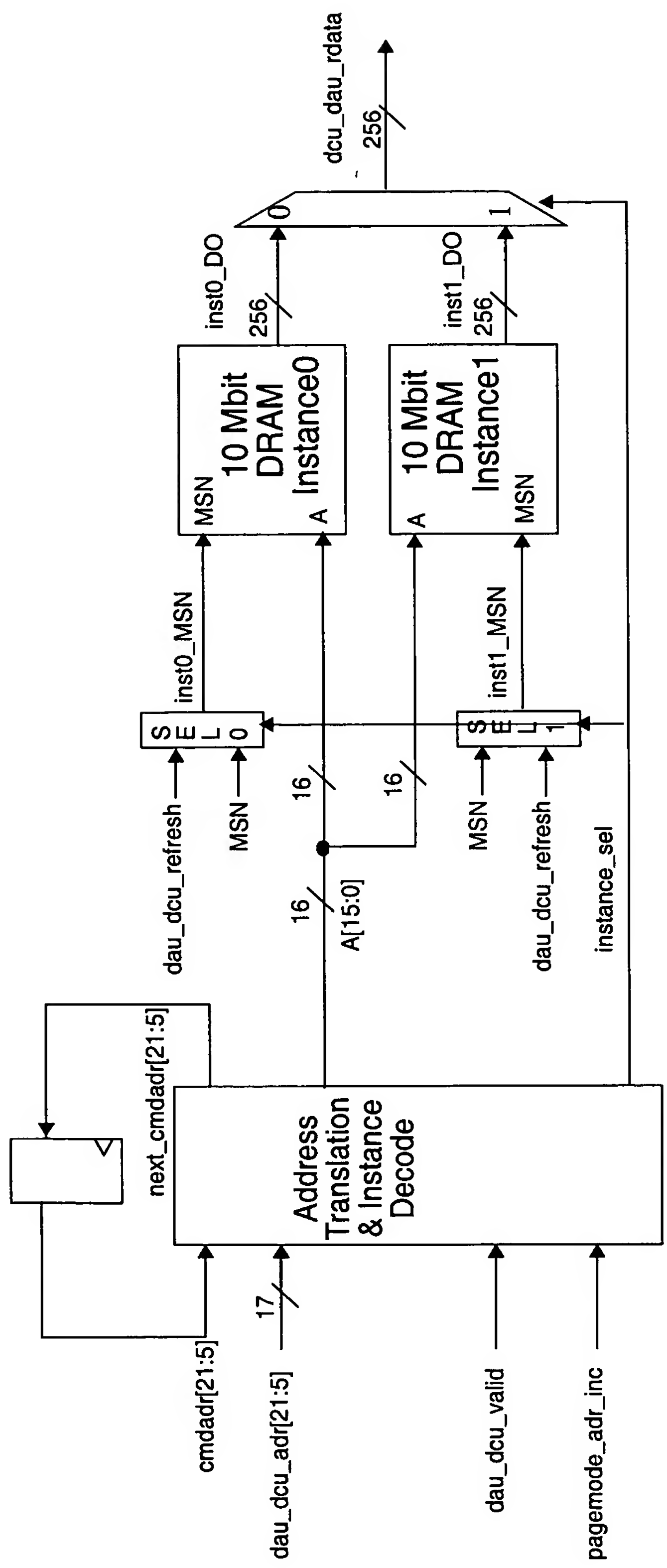


FIG. 102

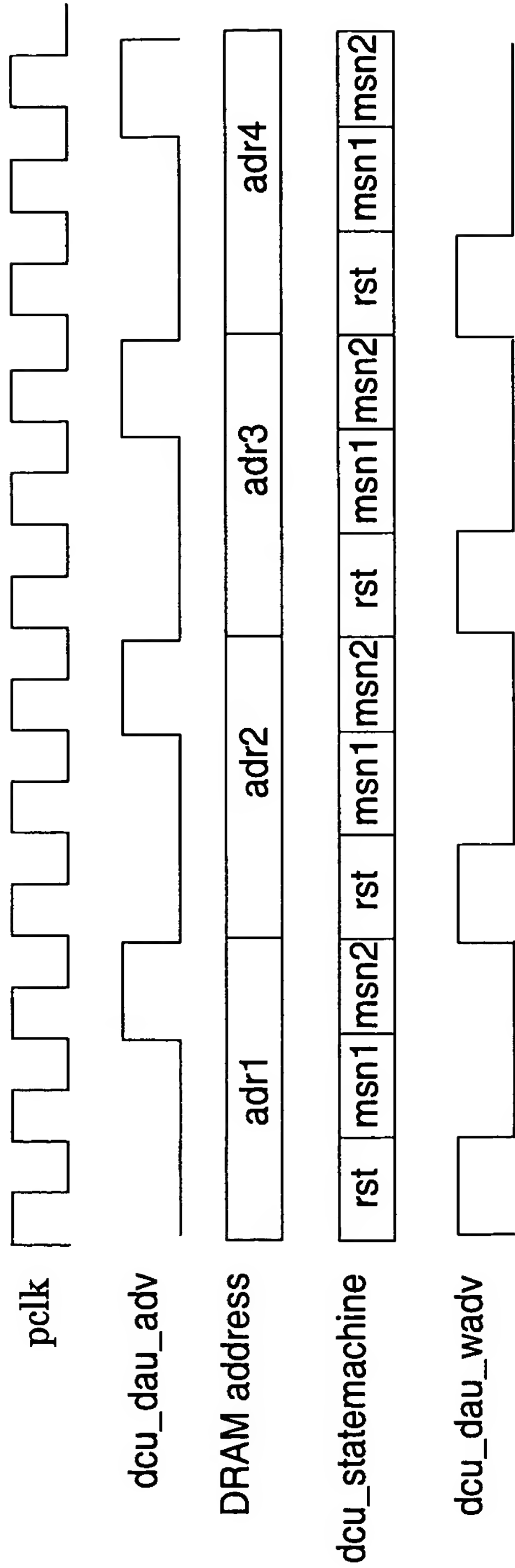


FIG. 103

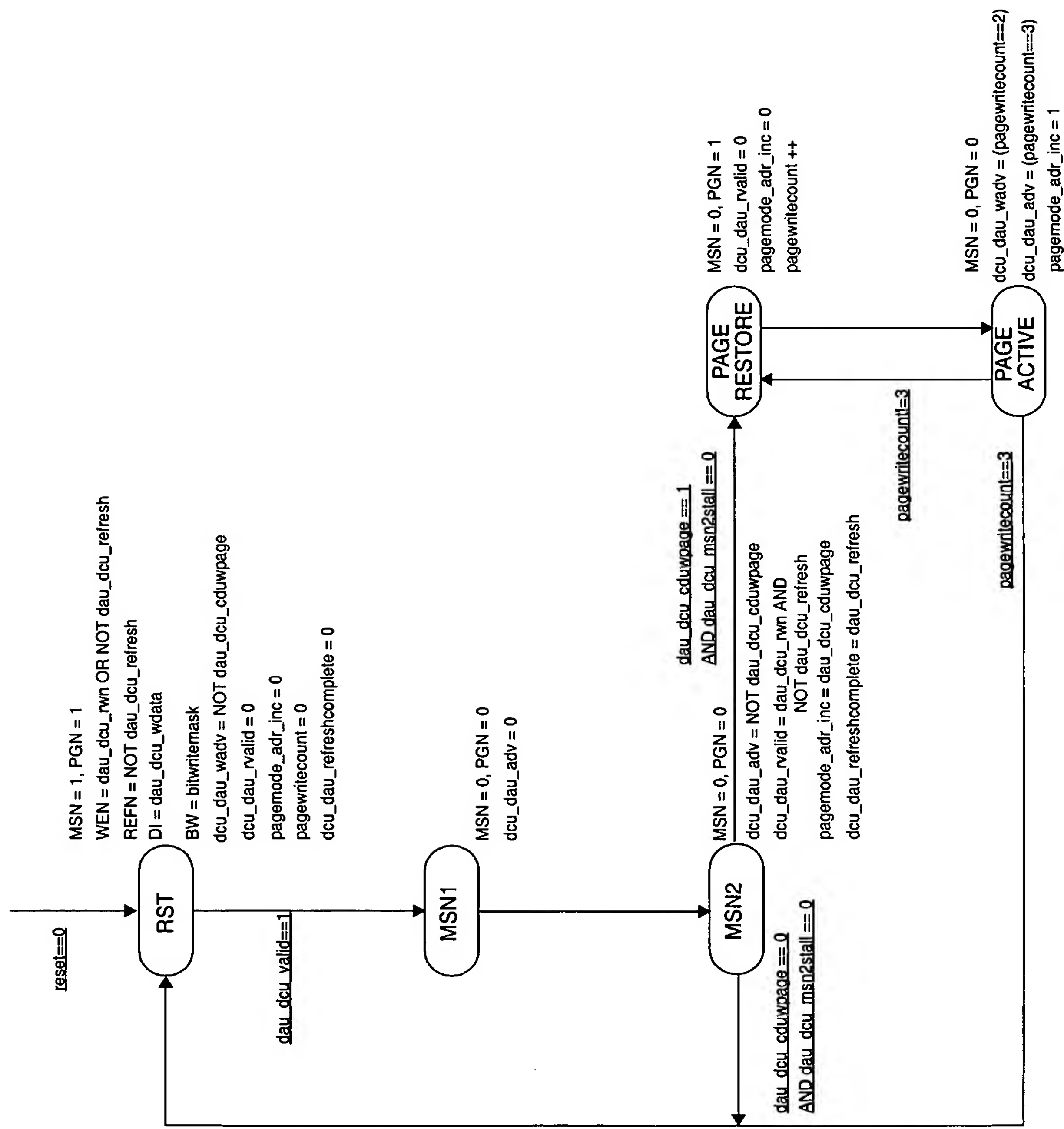


FIG. 104

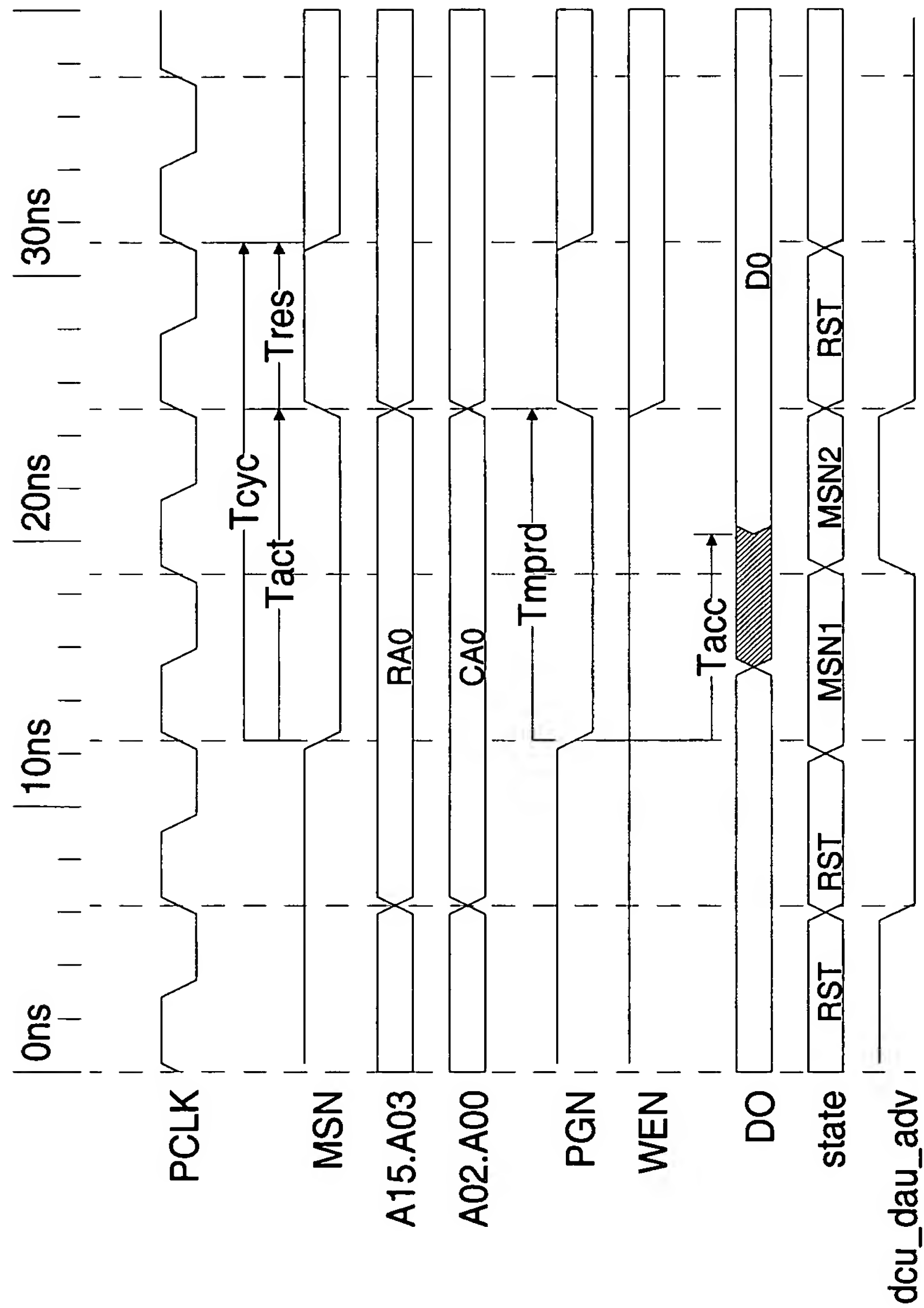


FIG. 105

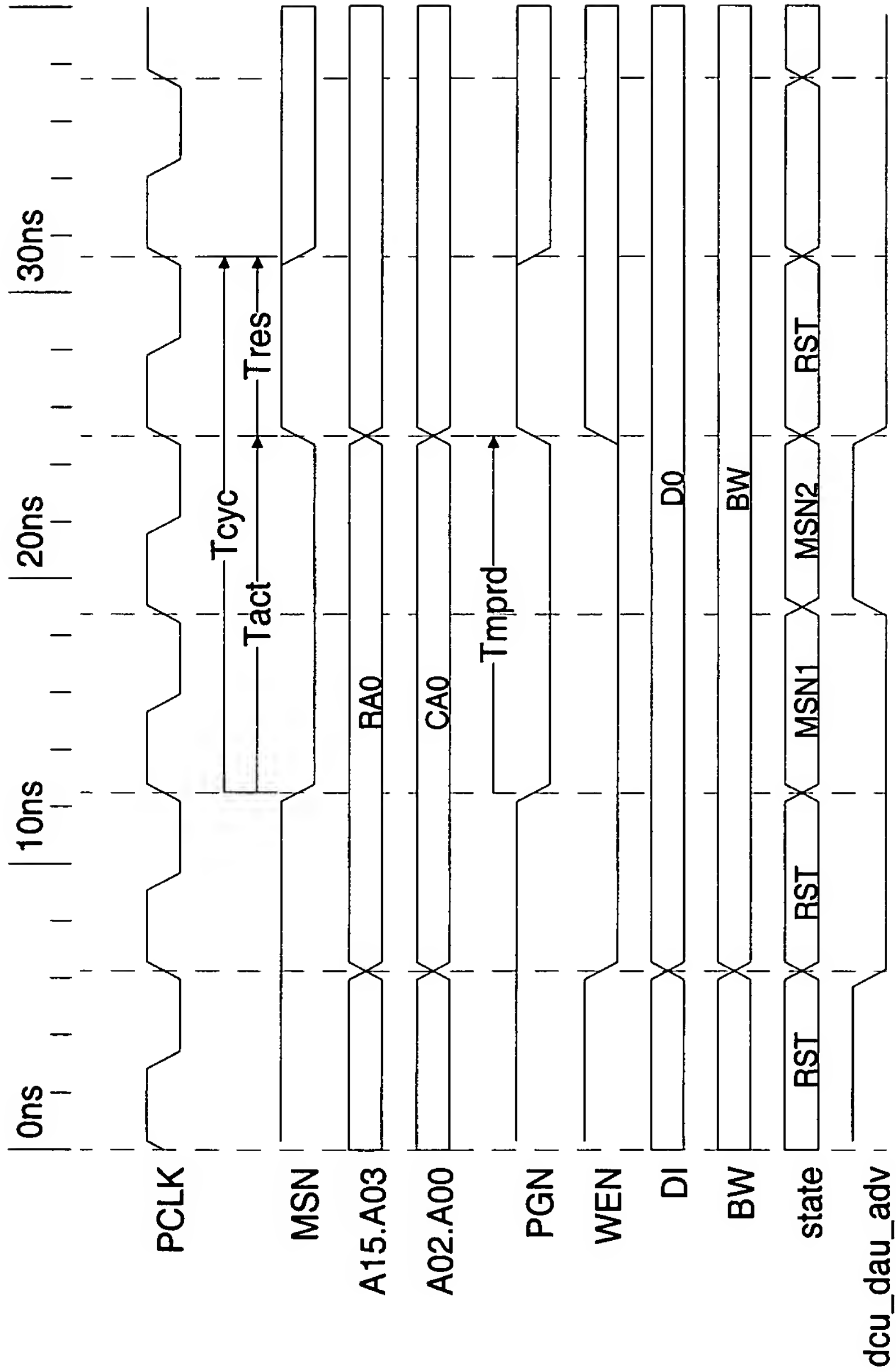


FIG. 106

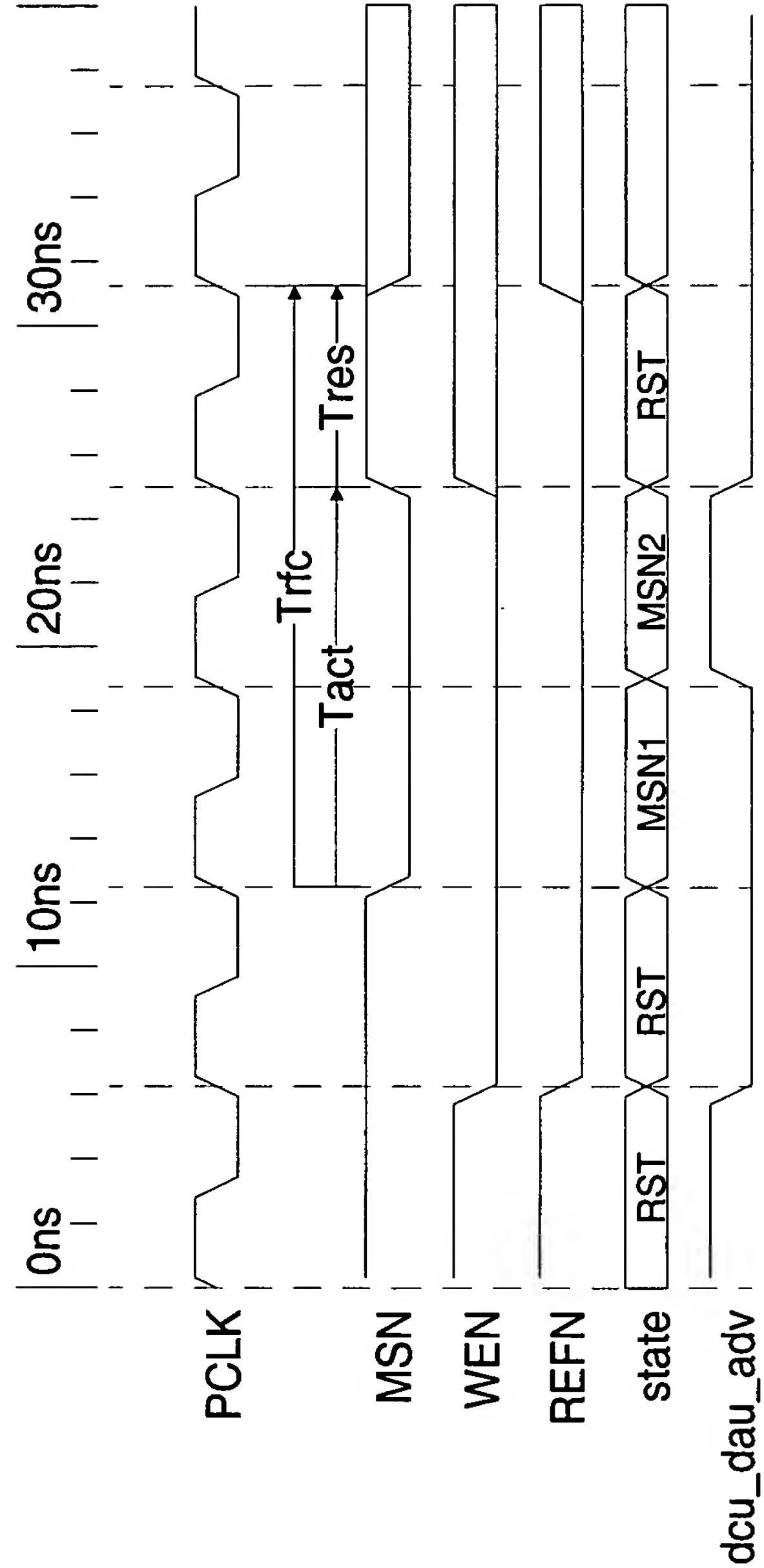


FIG. 107

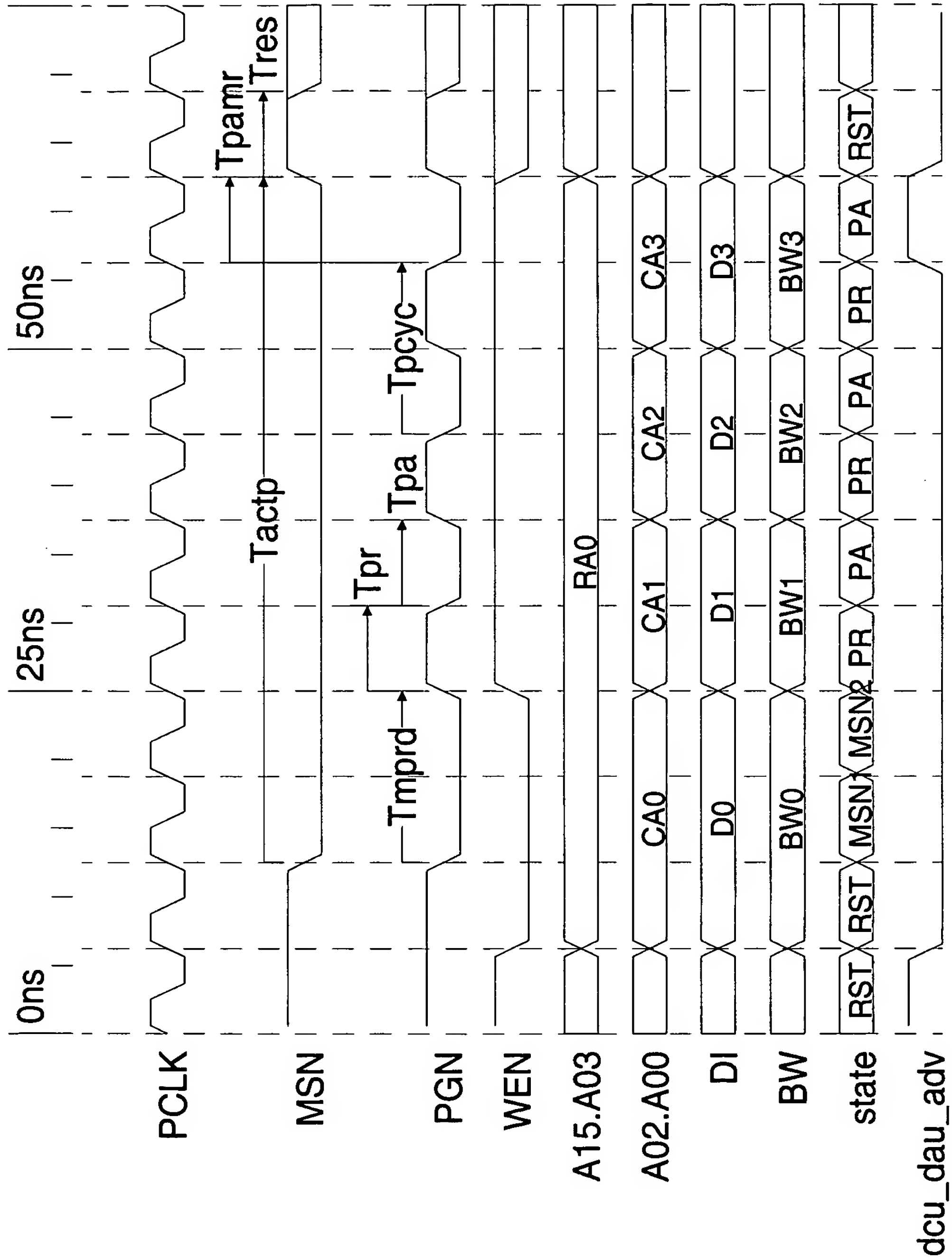


FIG. 108

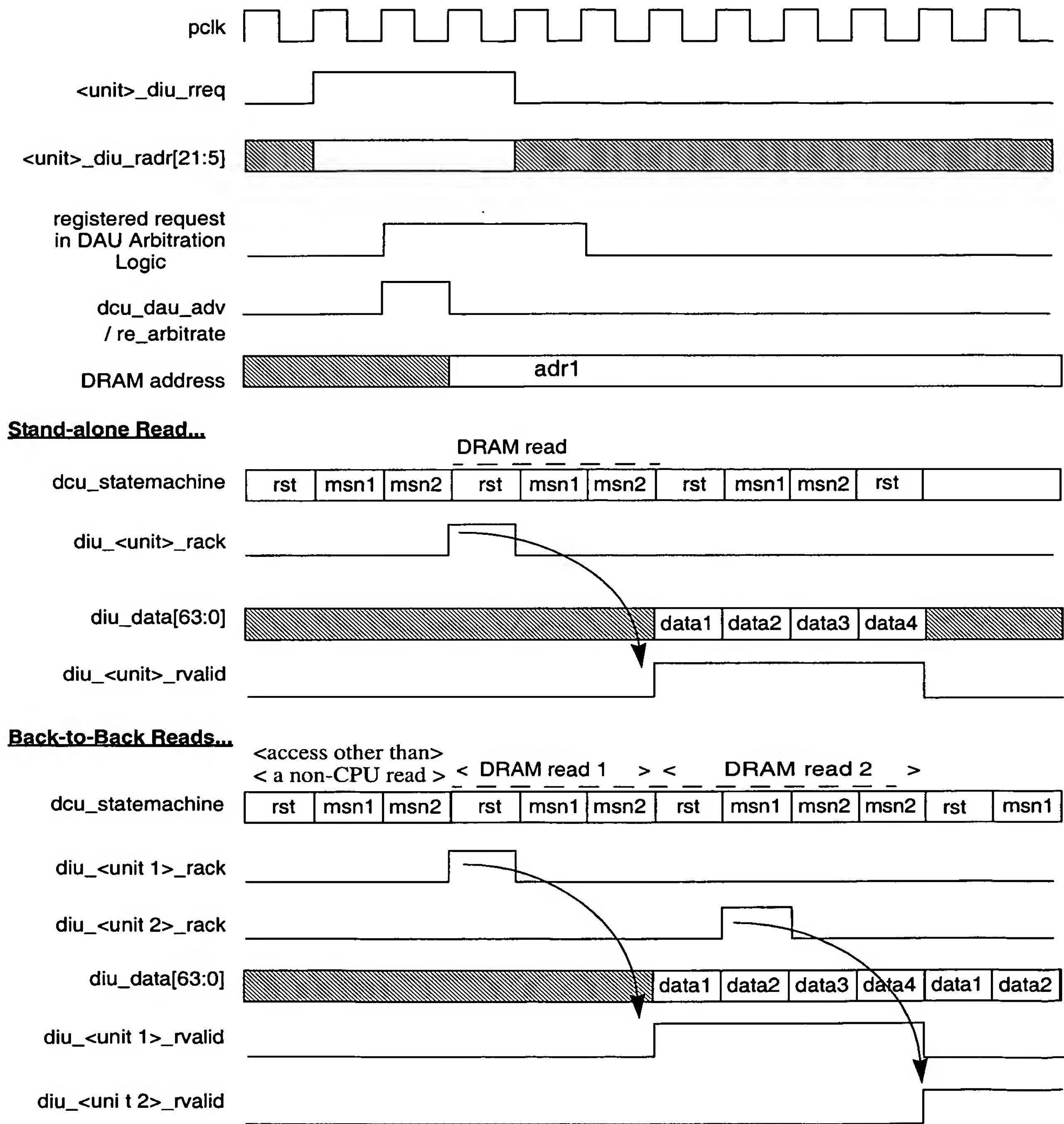


FIG. 109

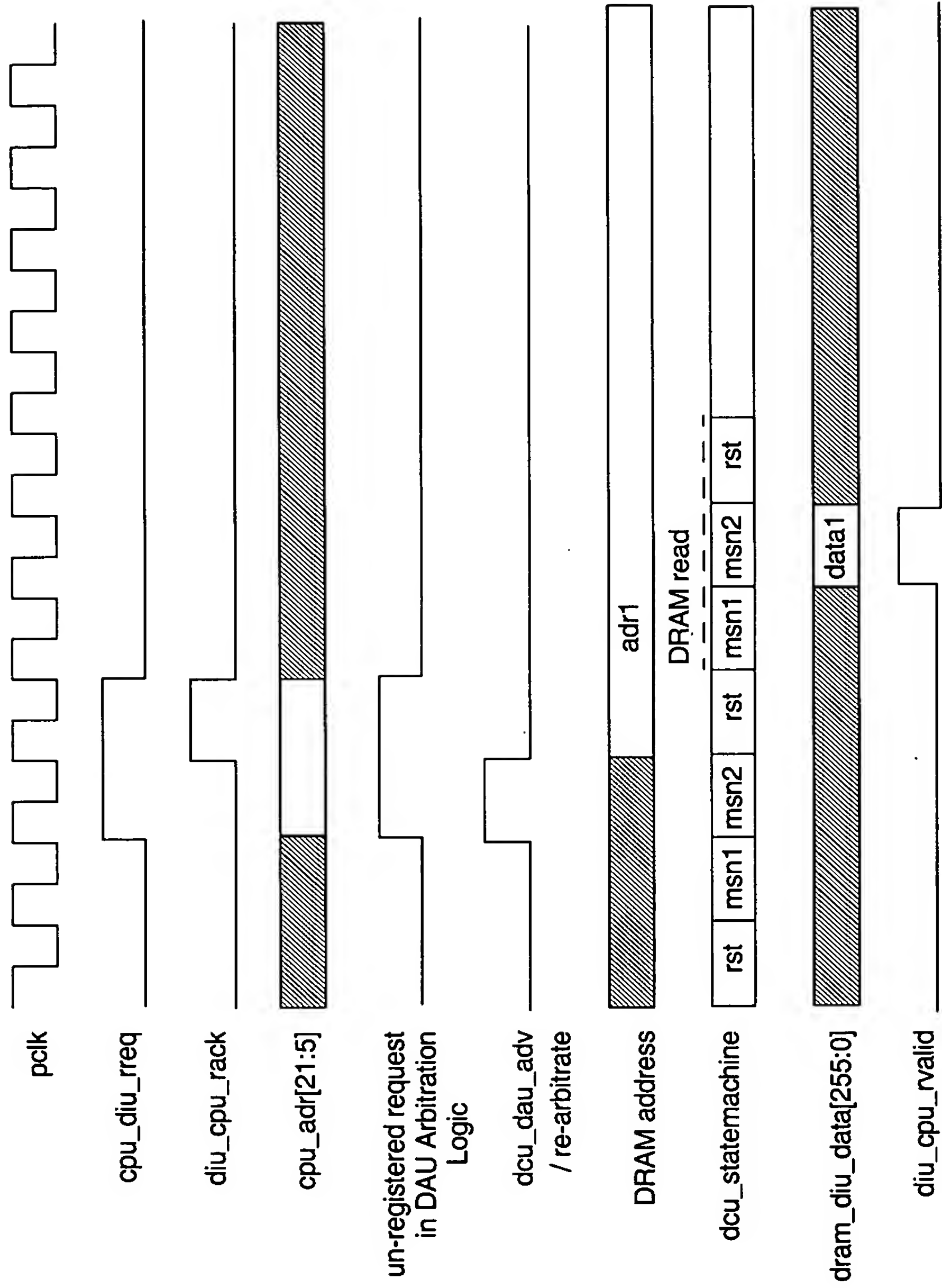


FIG. 110

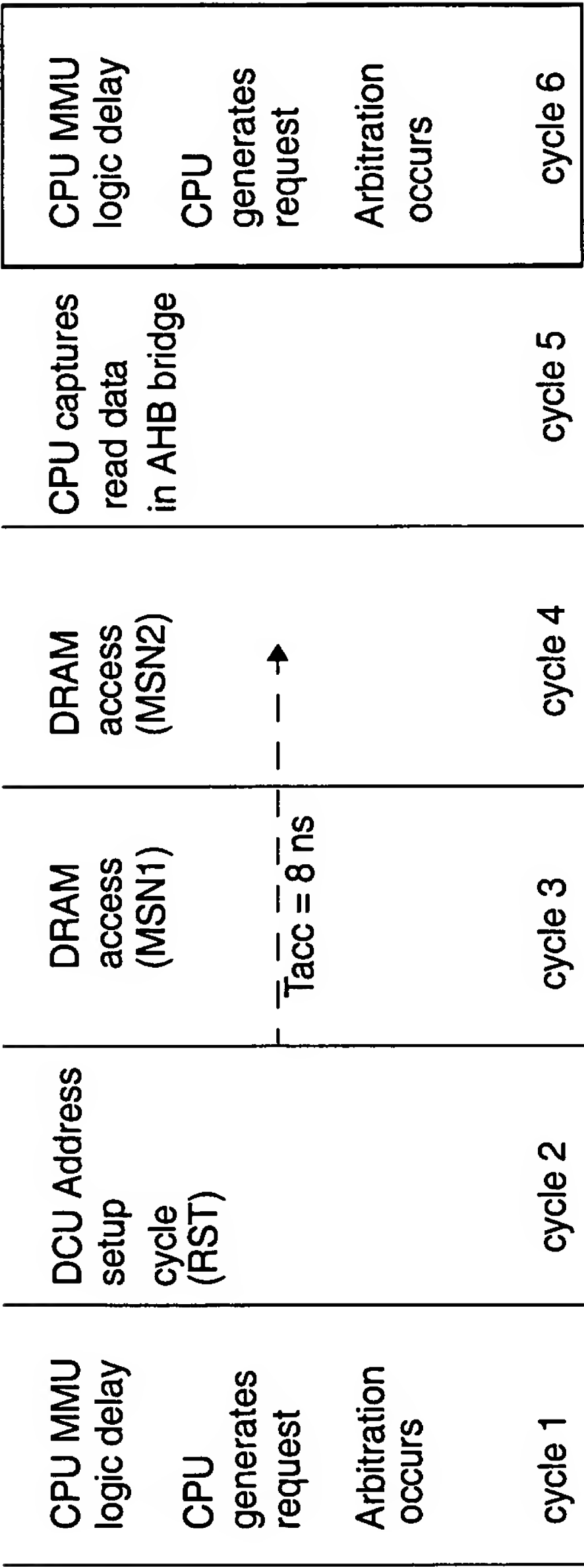


FIG. 111

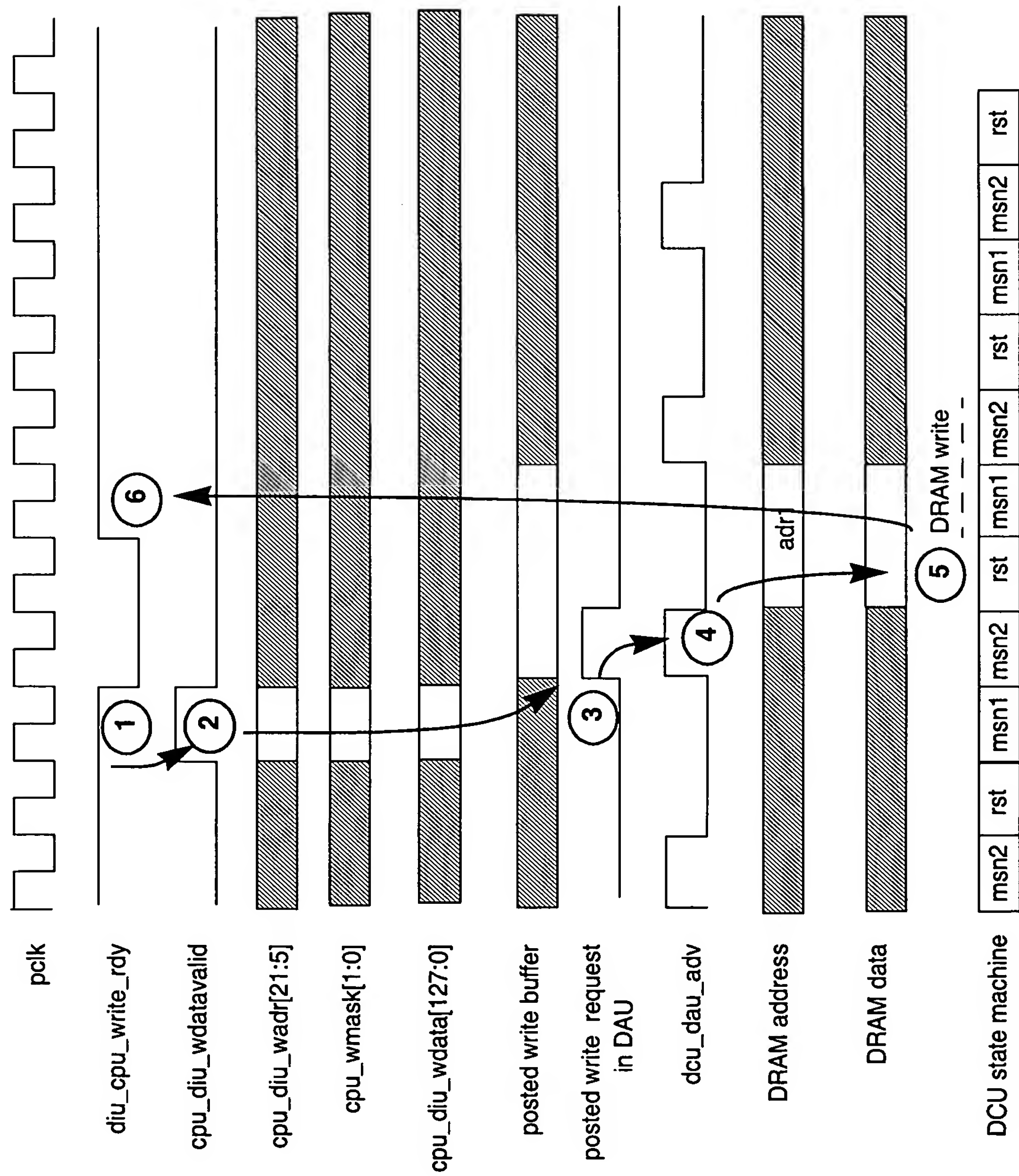


FIG. 112

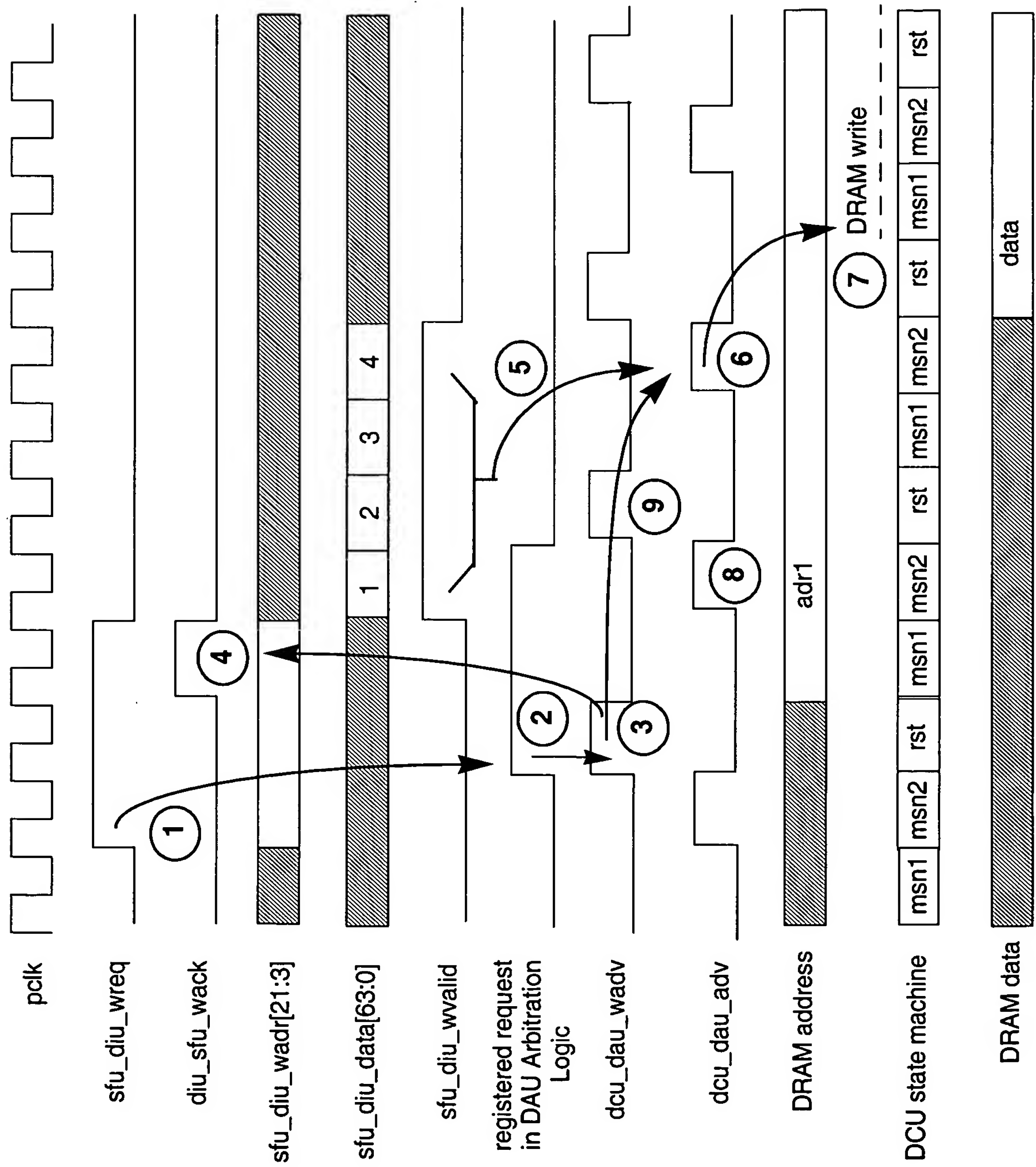


FIG. 113

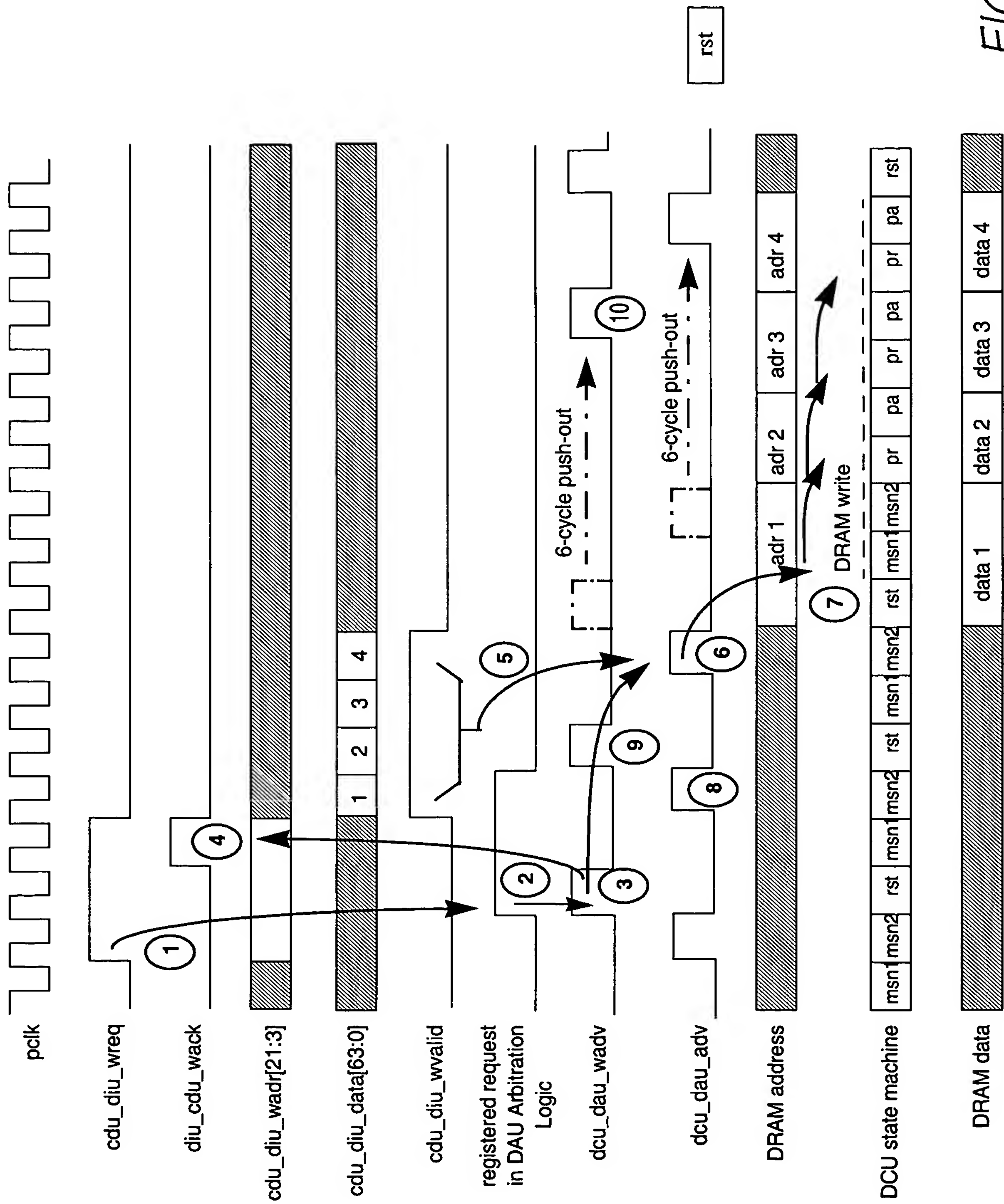


FIG. 114

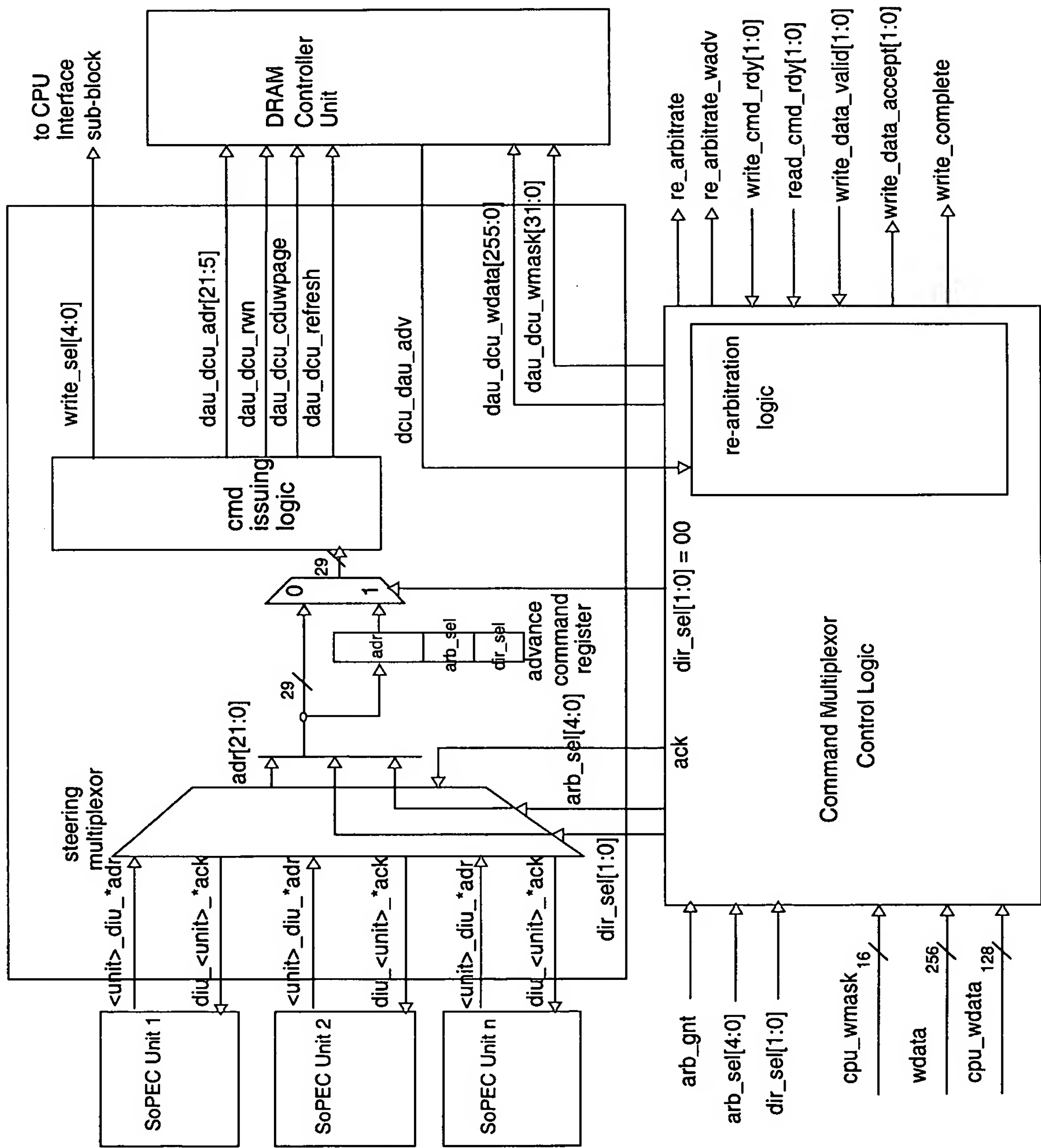


FIG. 115

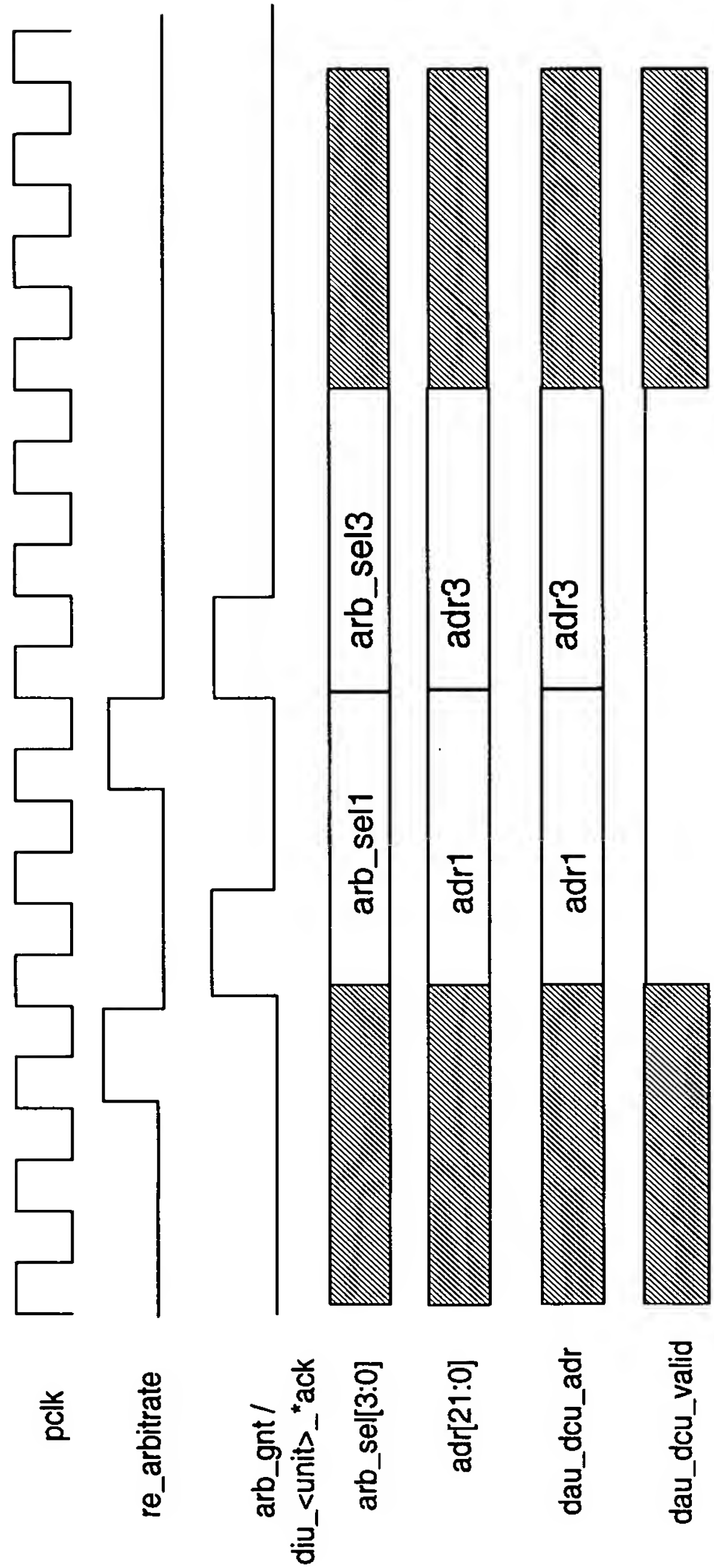


FIG. 116

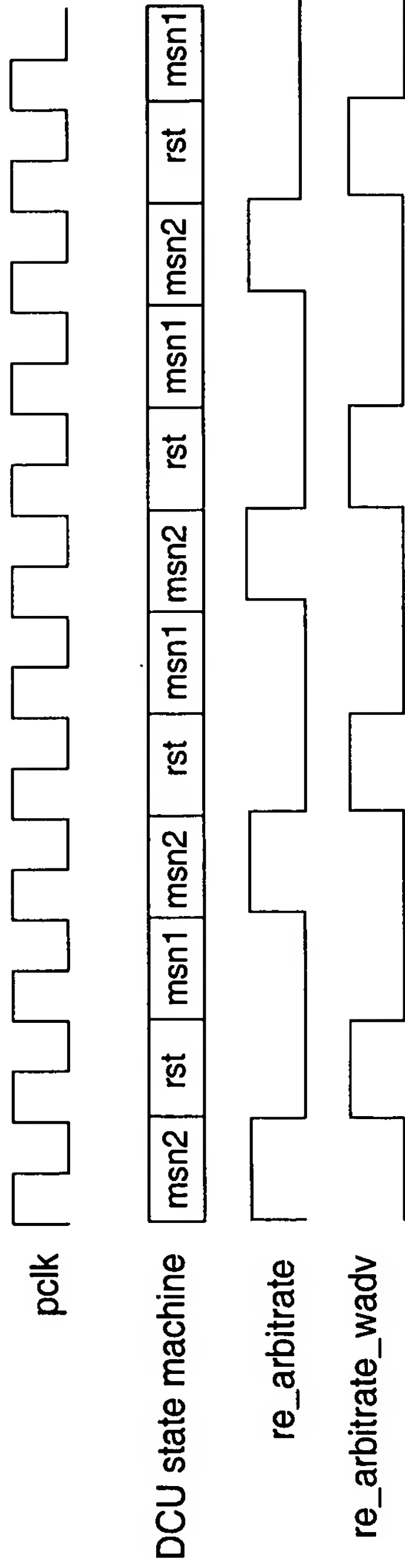


FIG. 117

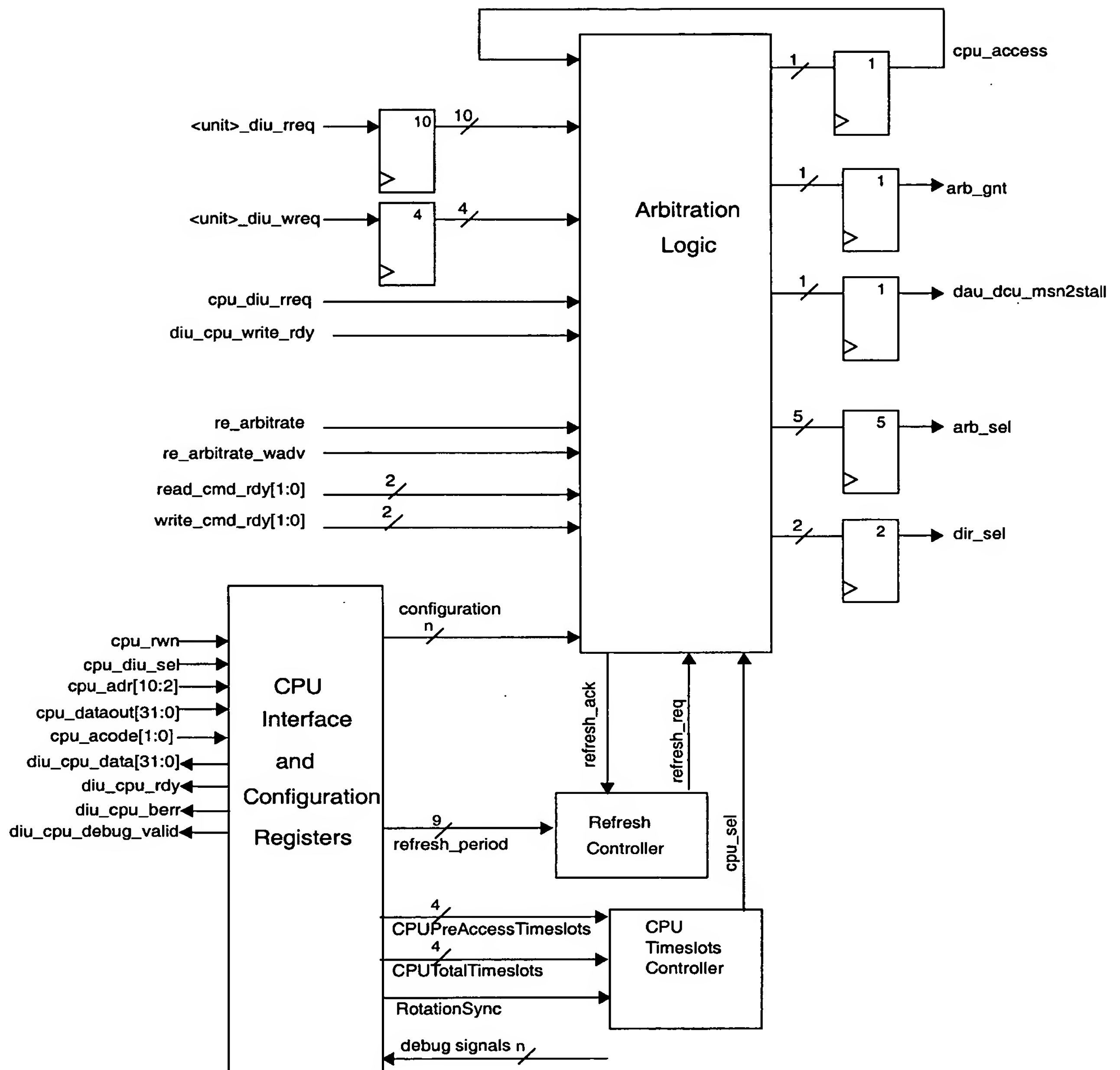


FIG. 118

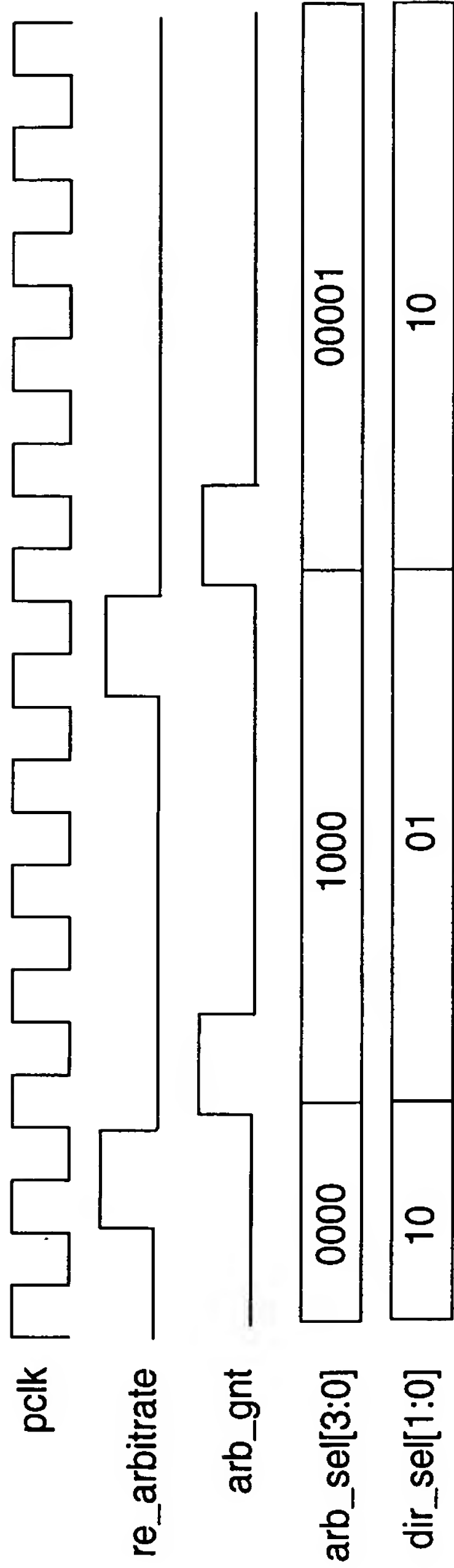


FIG. 119

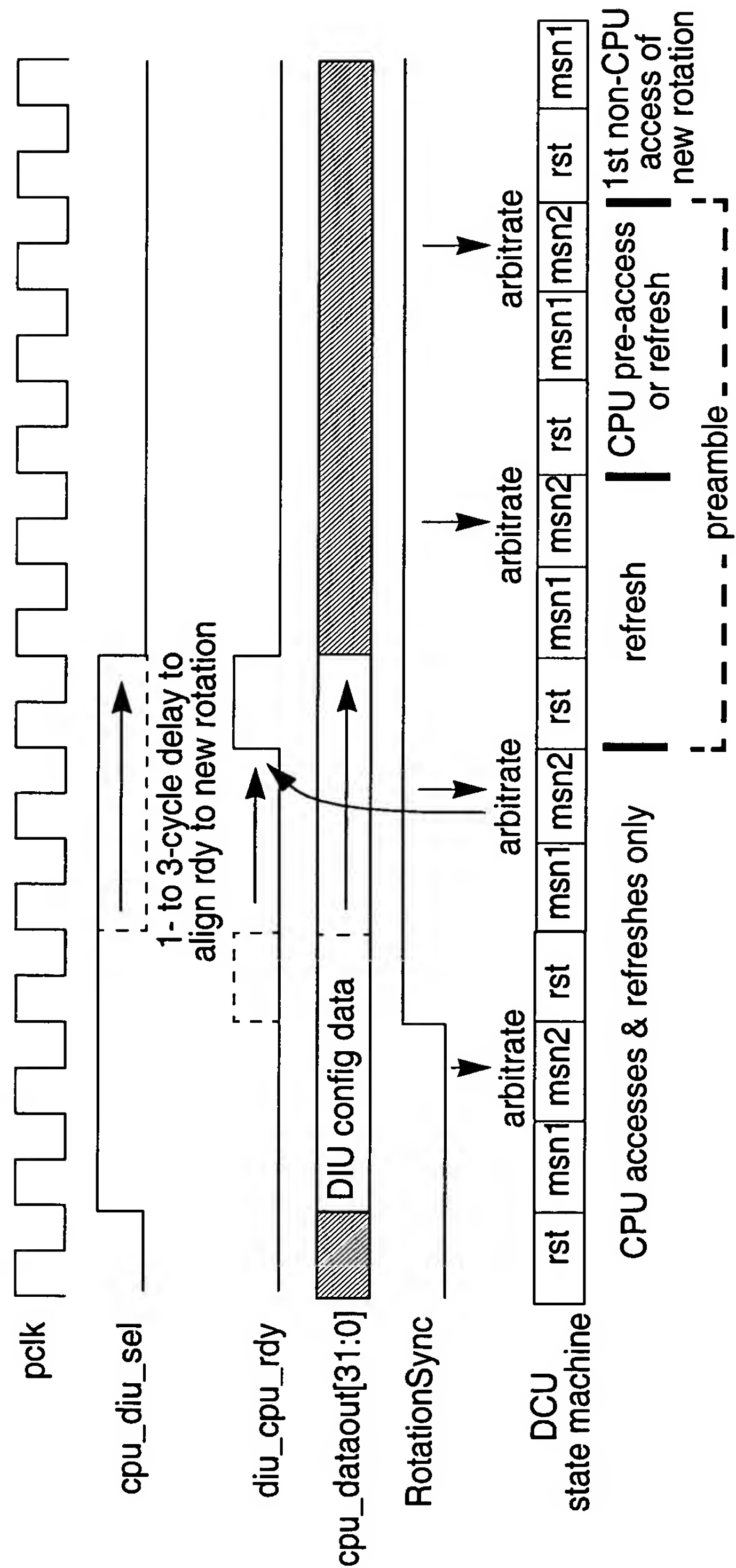


FIG. 120

106/331

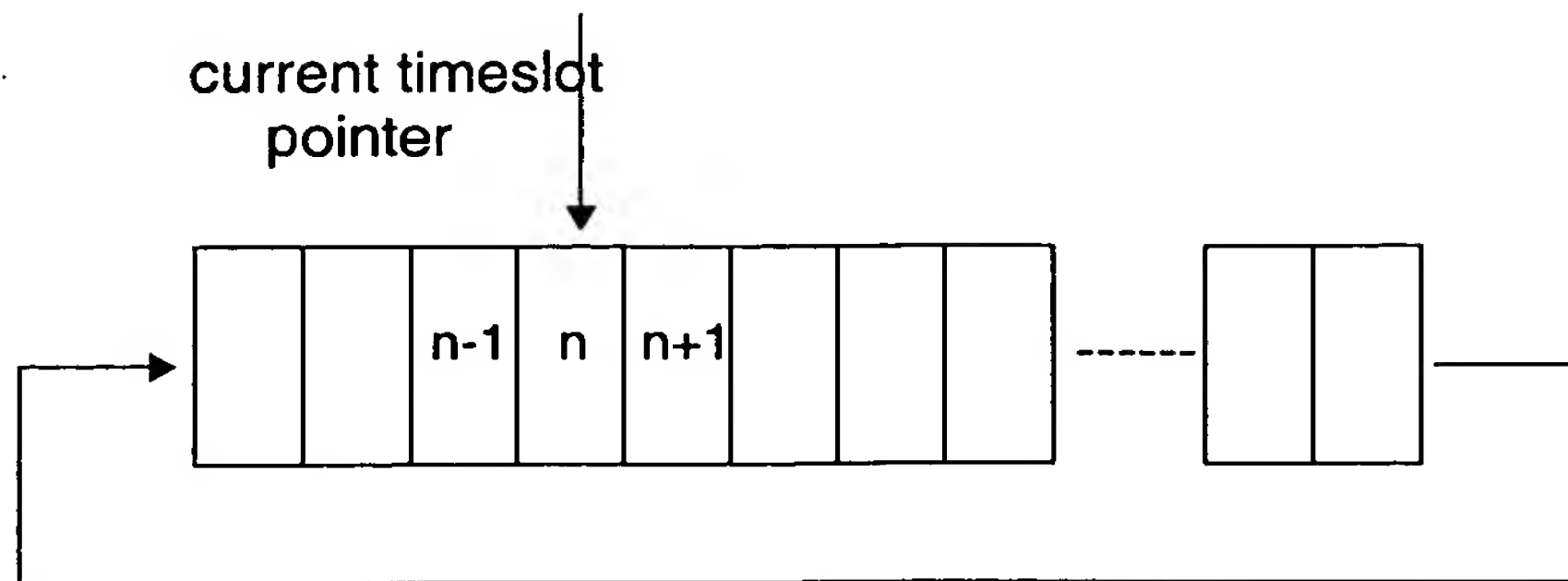


FIG. 121

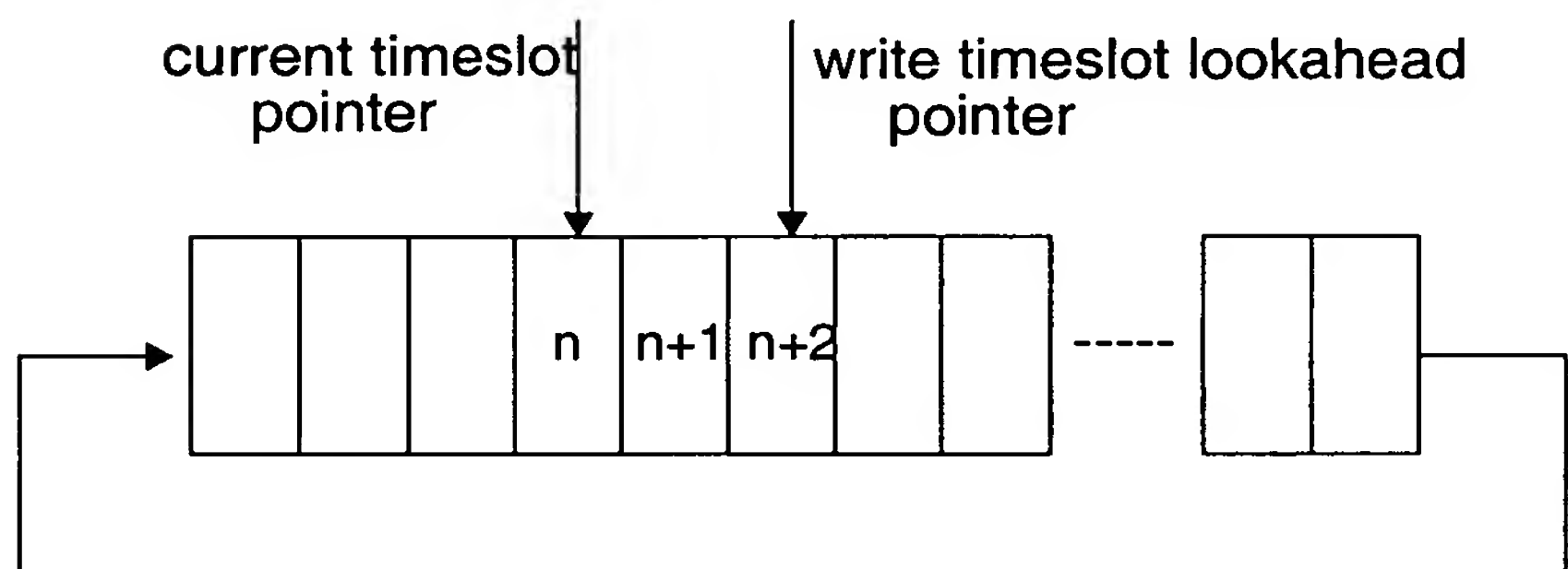


FIG. 122

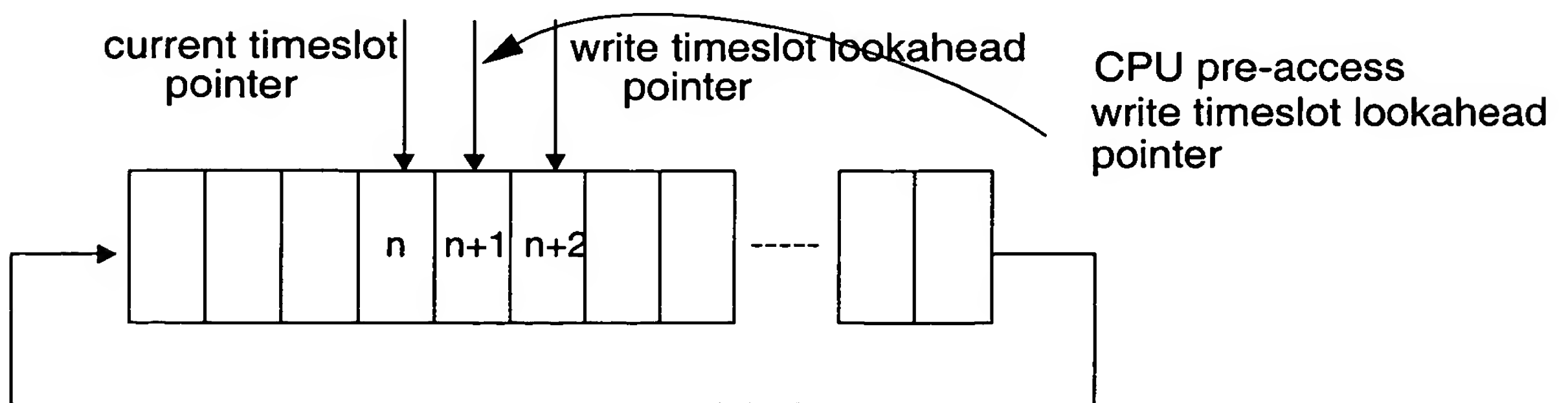


FIG. 123

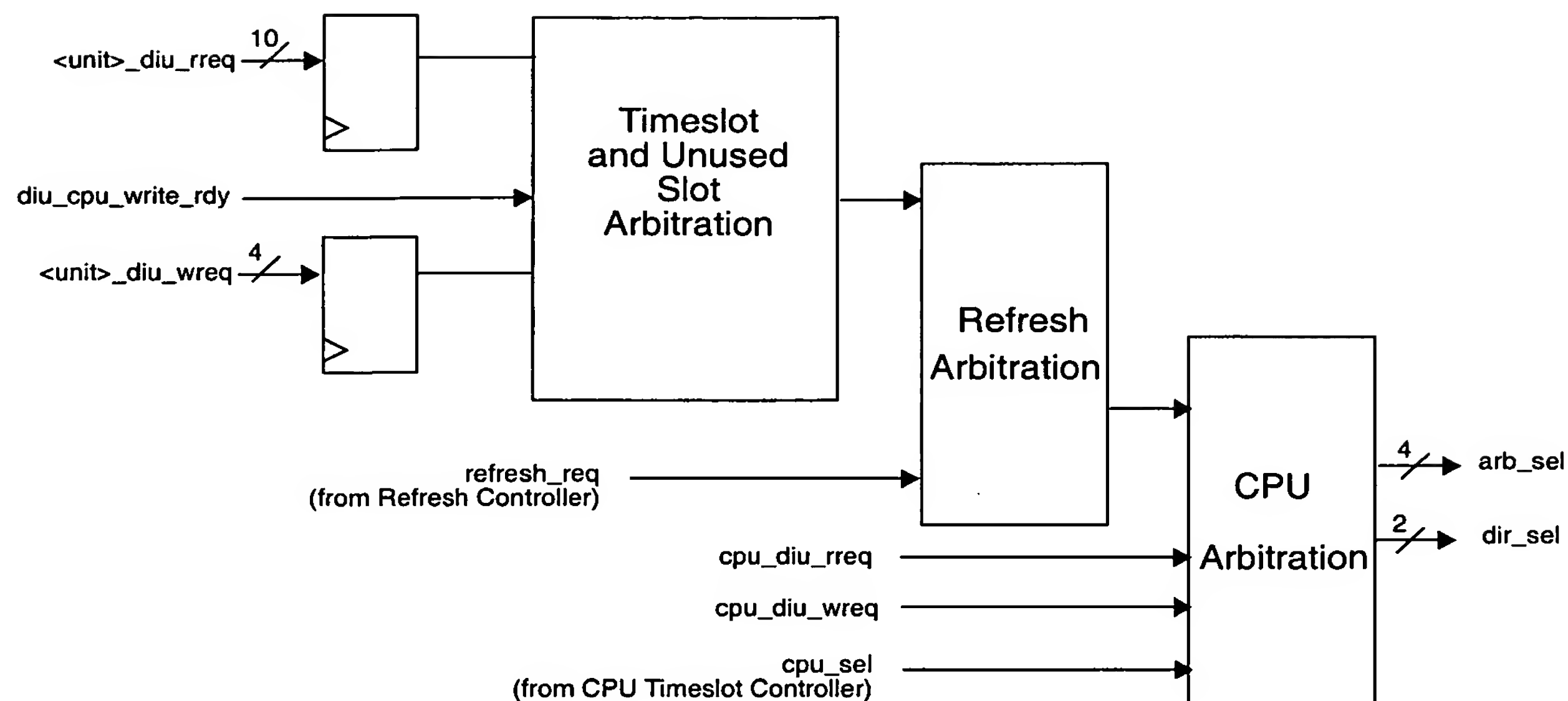


FIG. 124

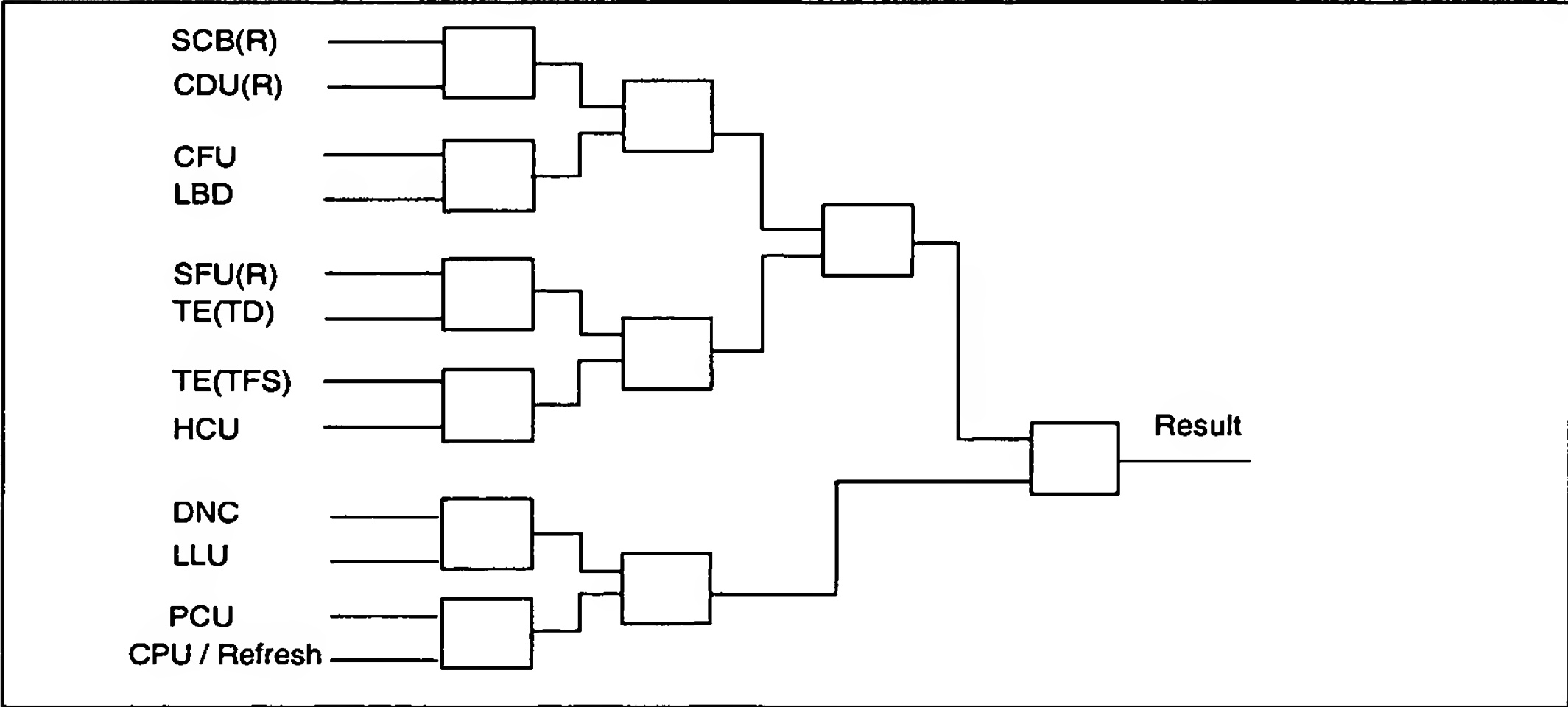


FIG. 125

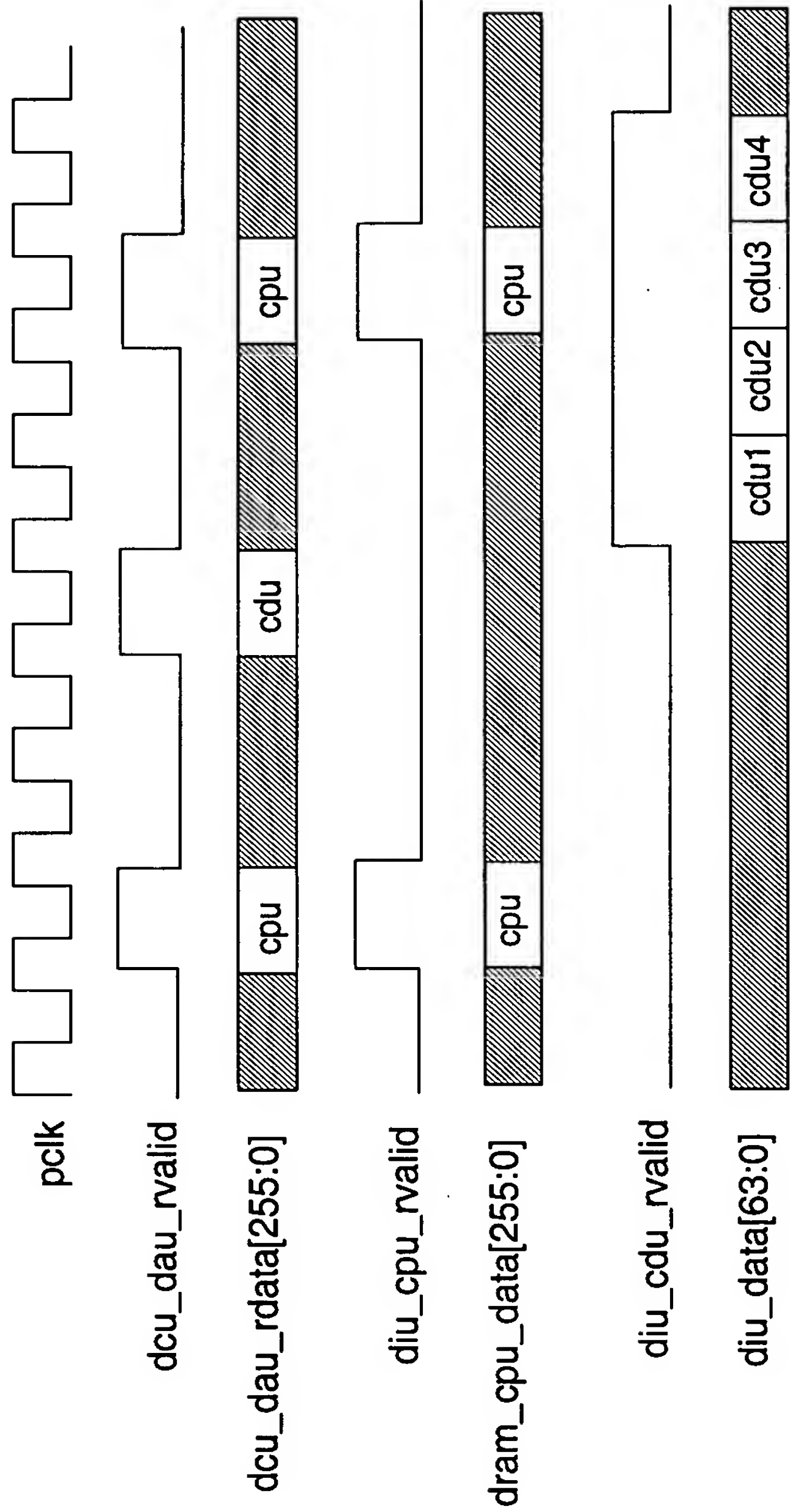


FIG. 127

110/331

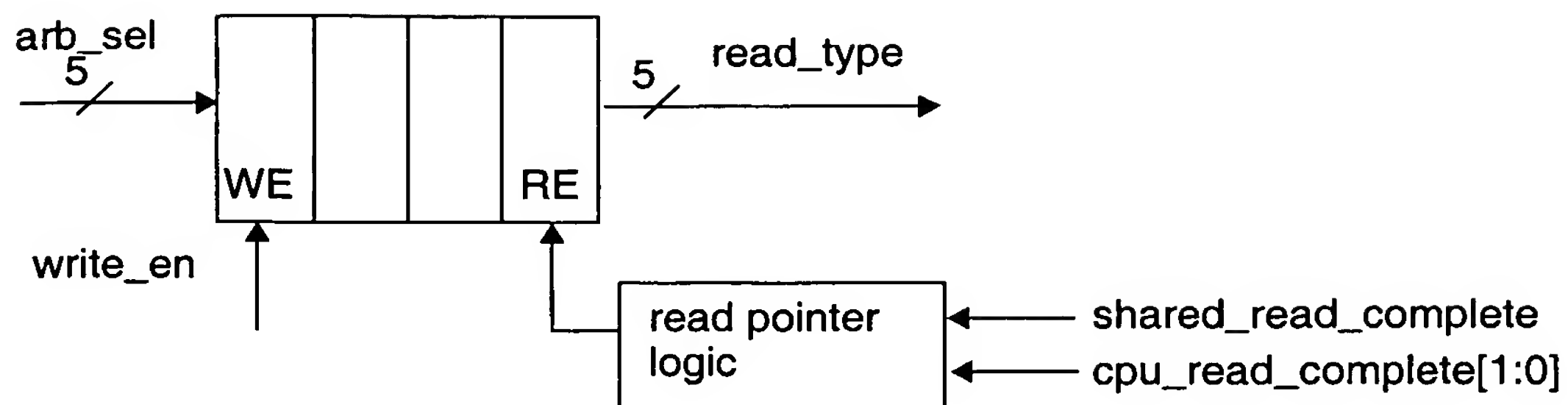


FIG. 128

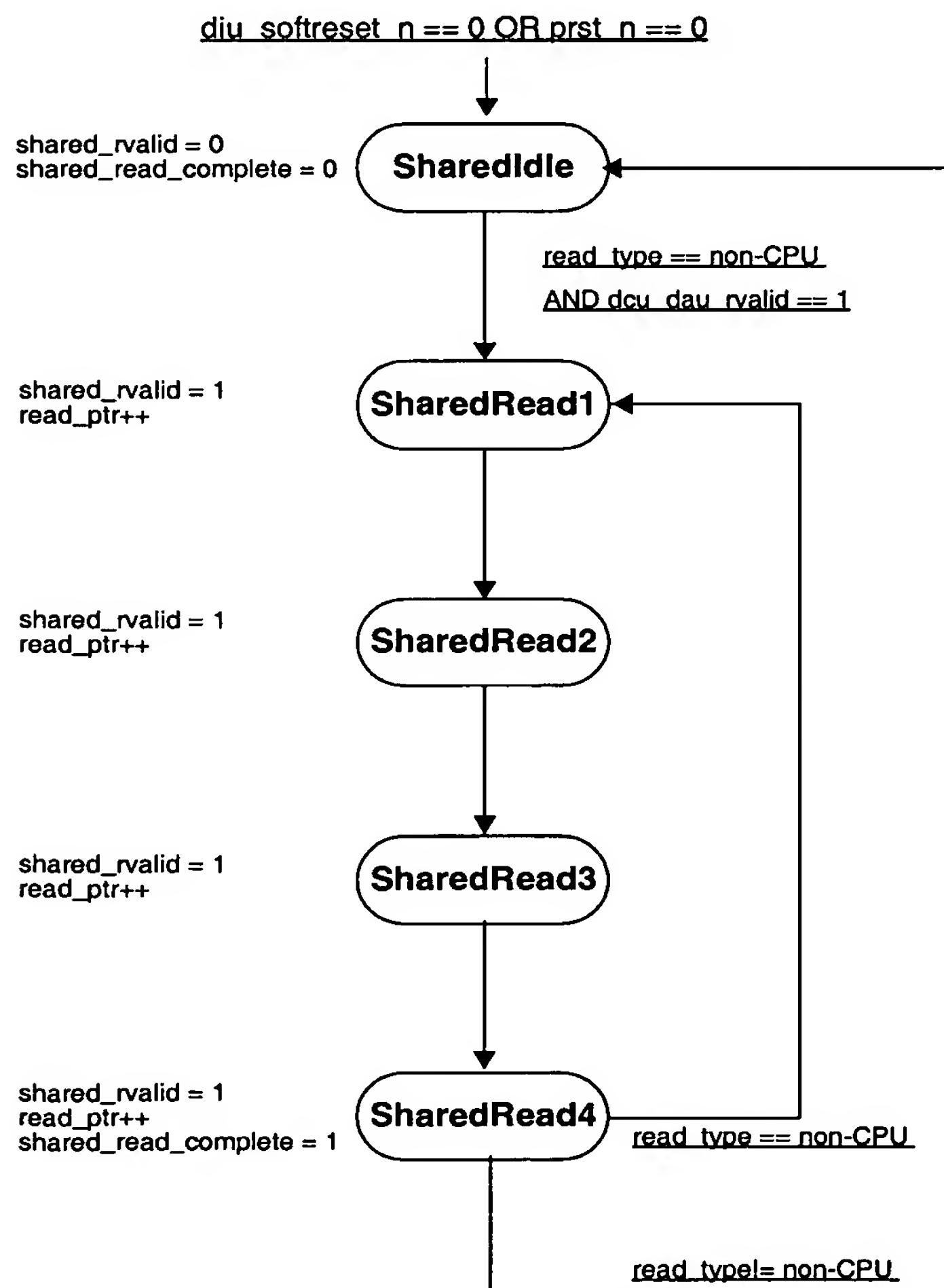


FIG. 129

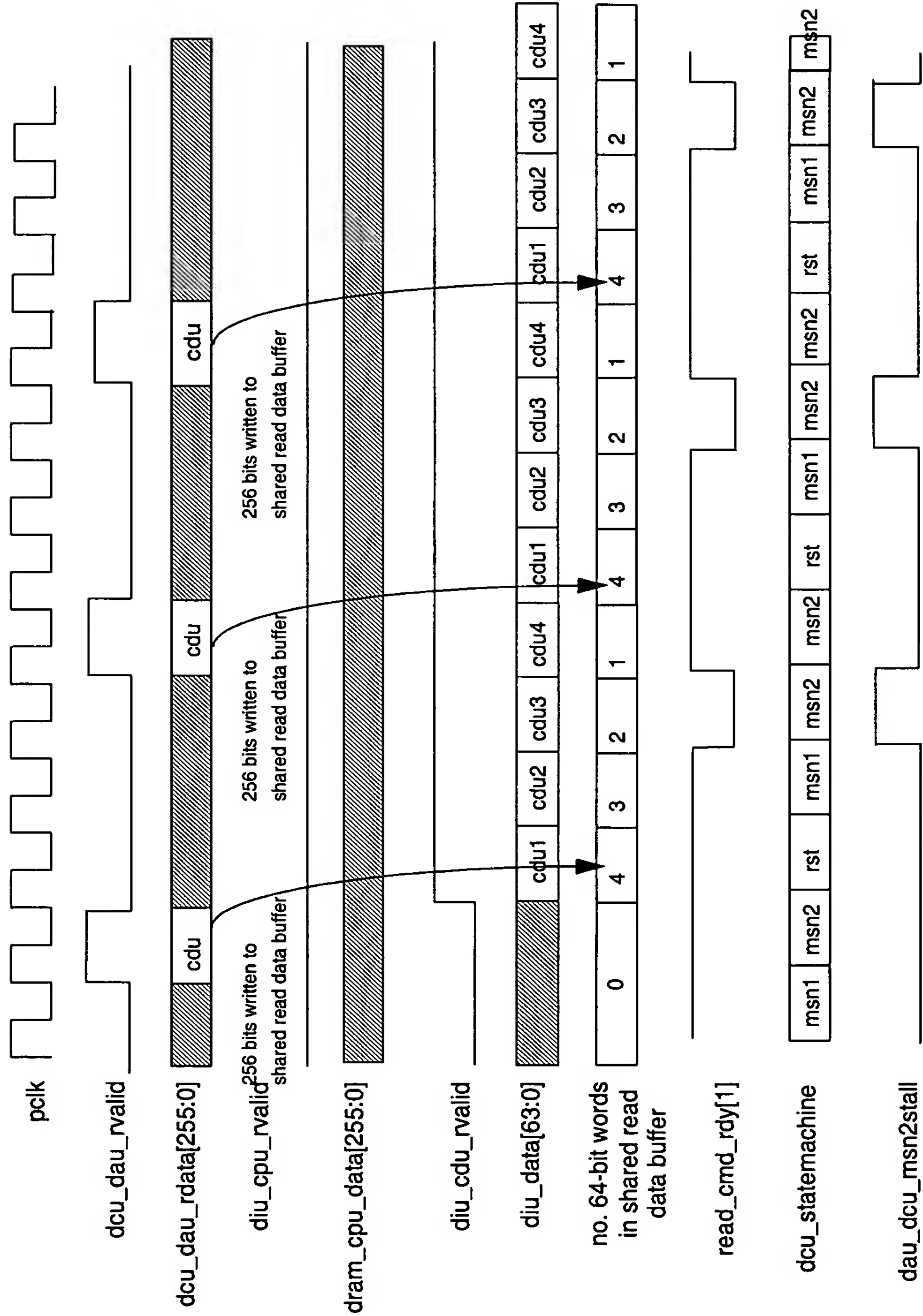


FIG. 130

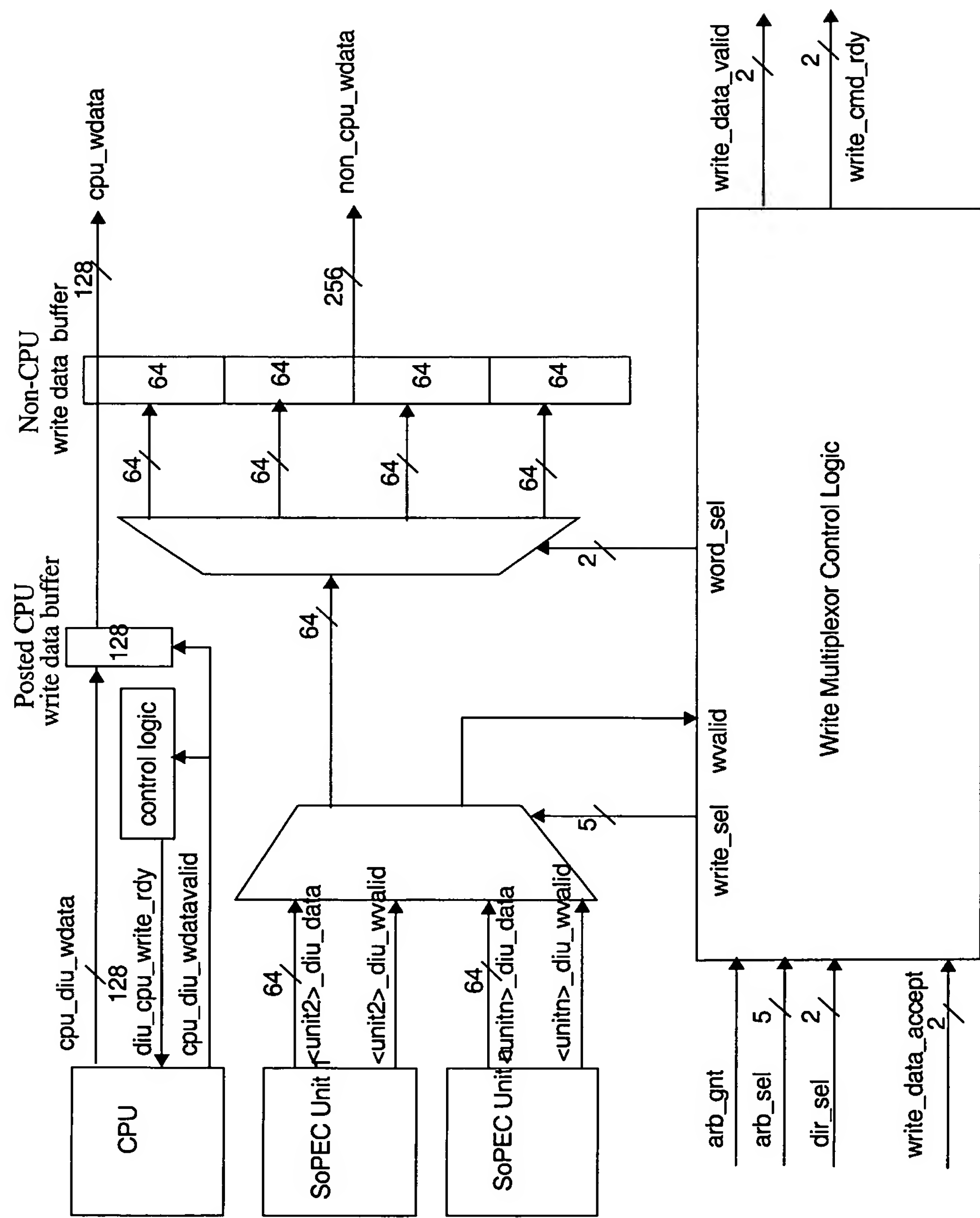


FIG. 131

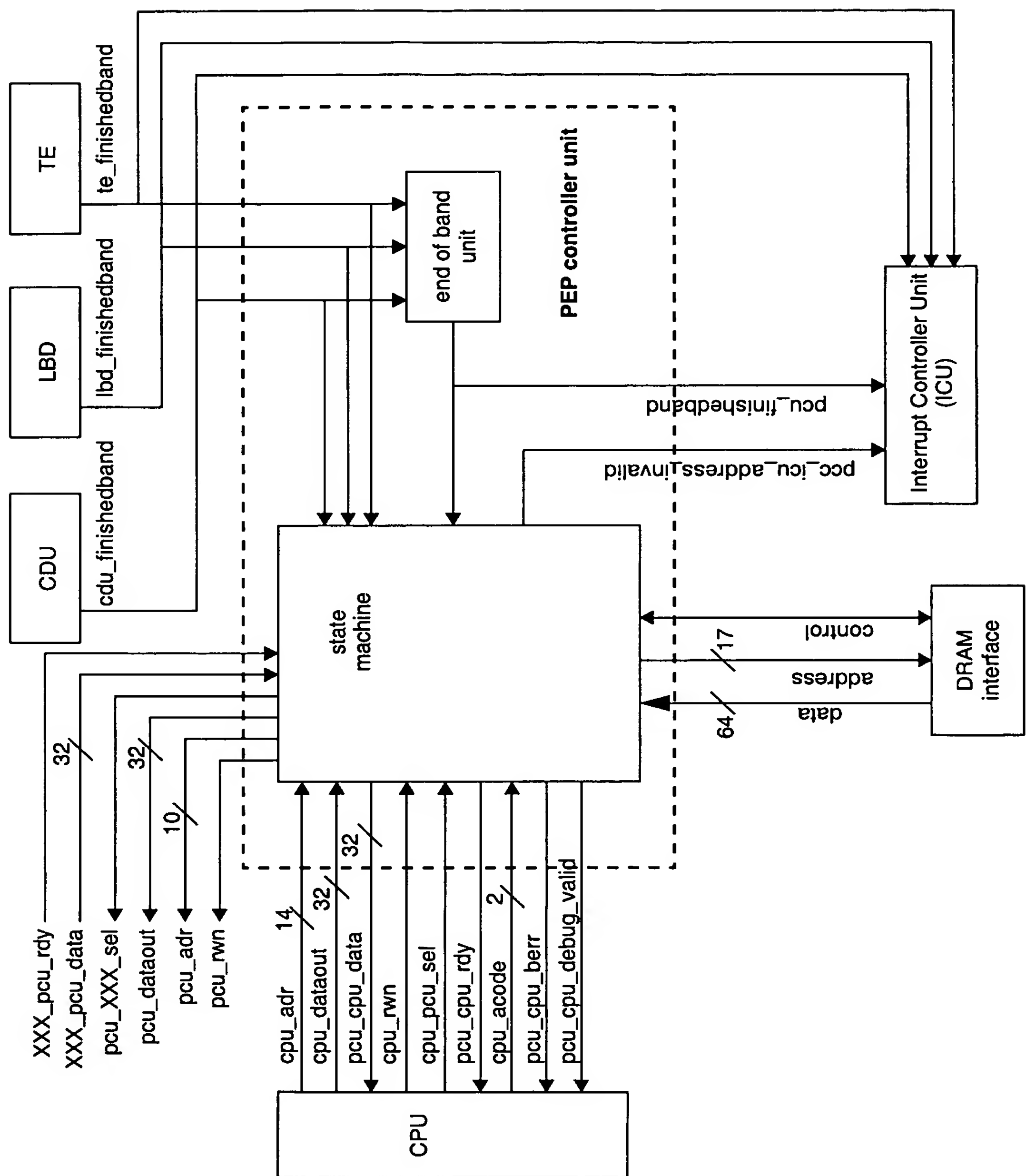


FIG. 132

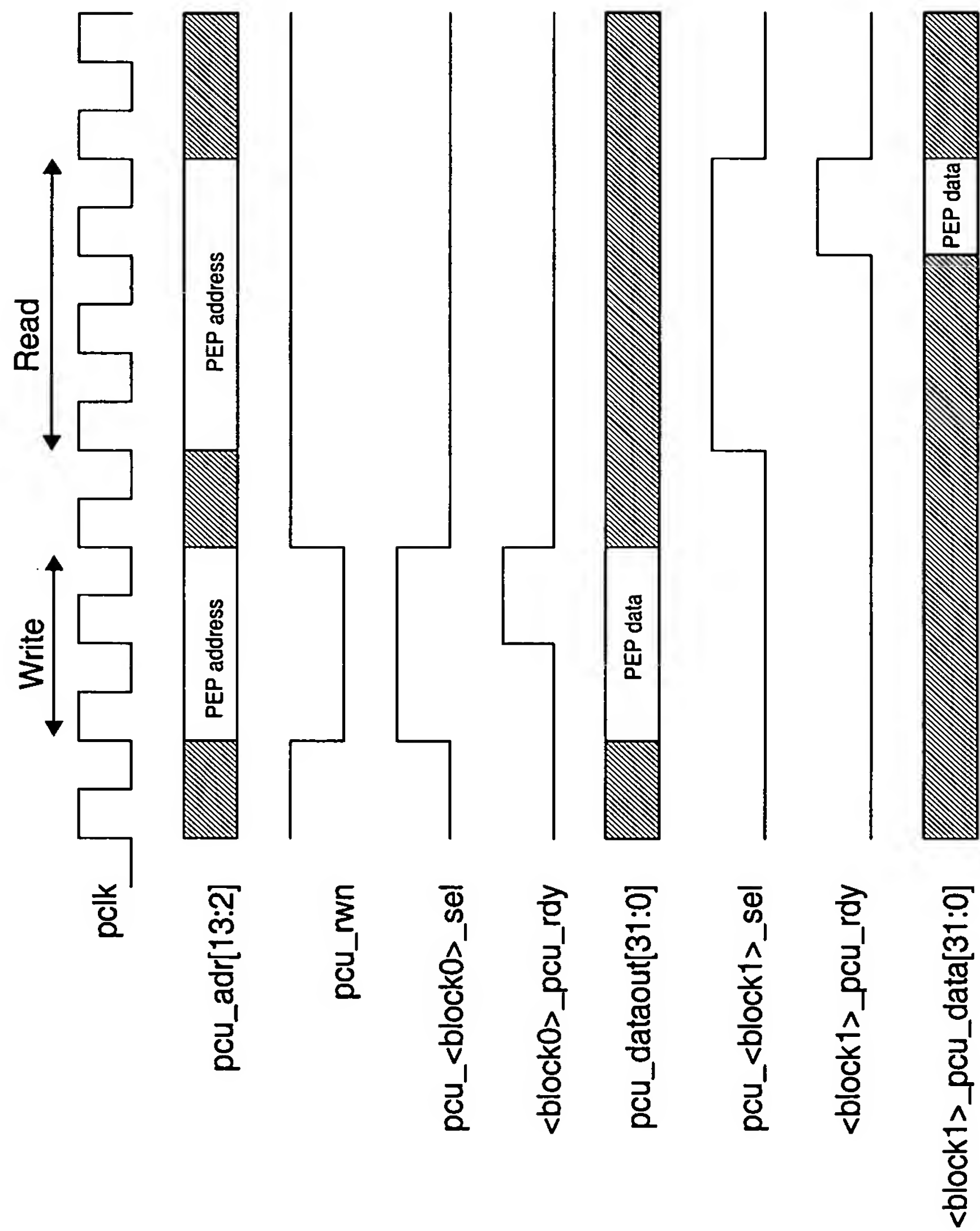


FIG. 133

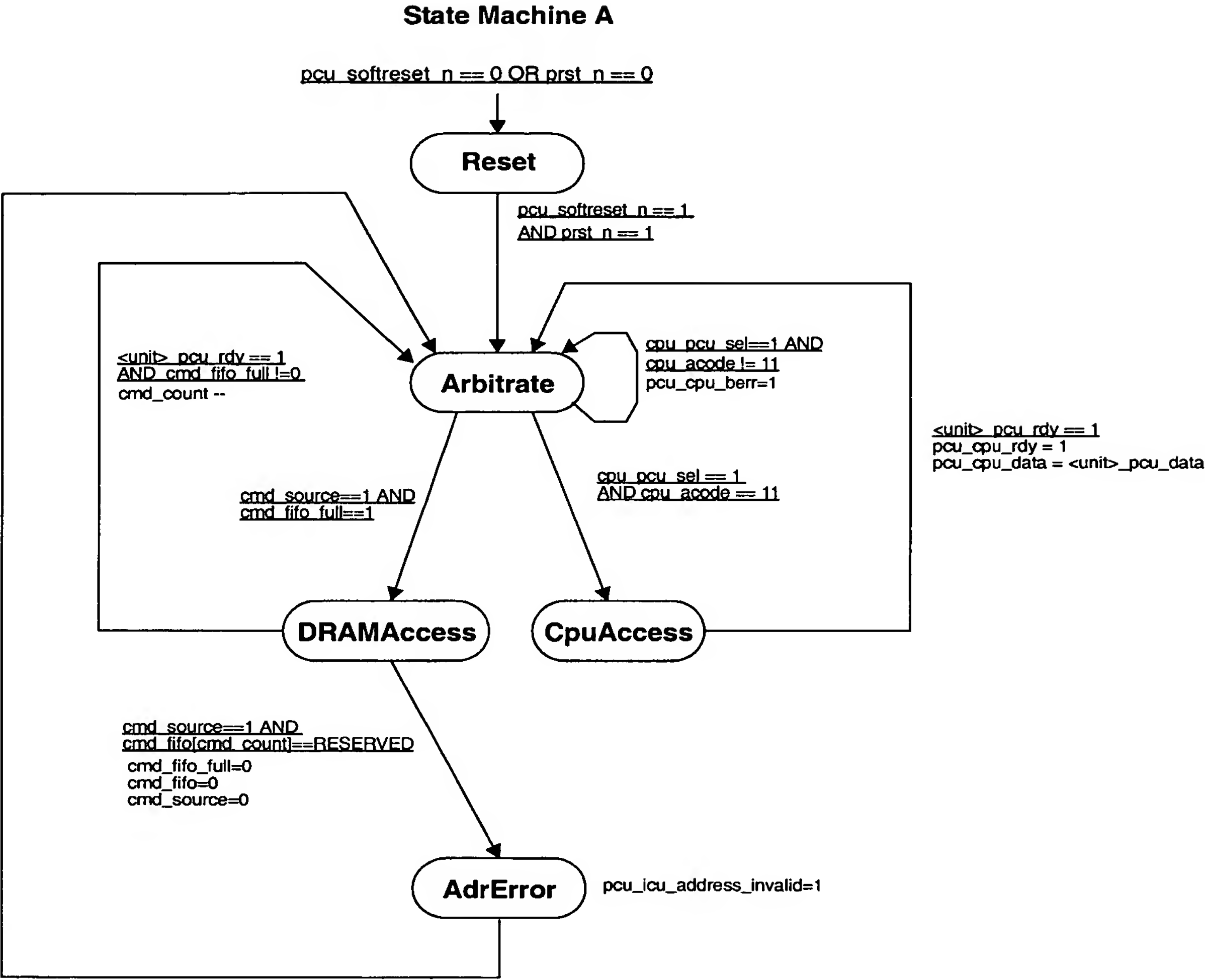


FIG. 134

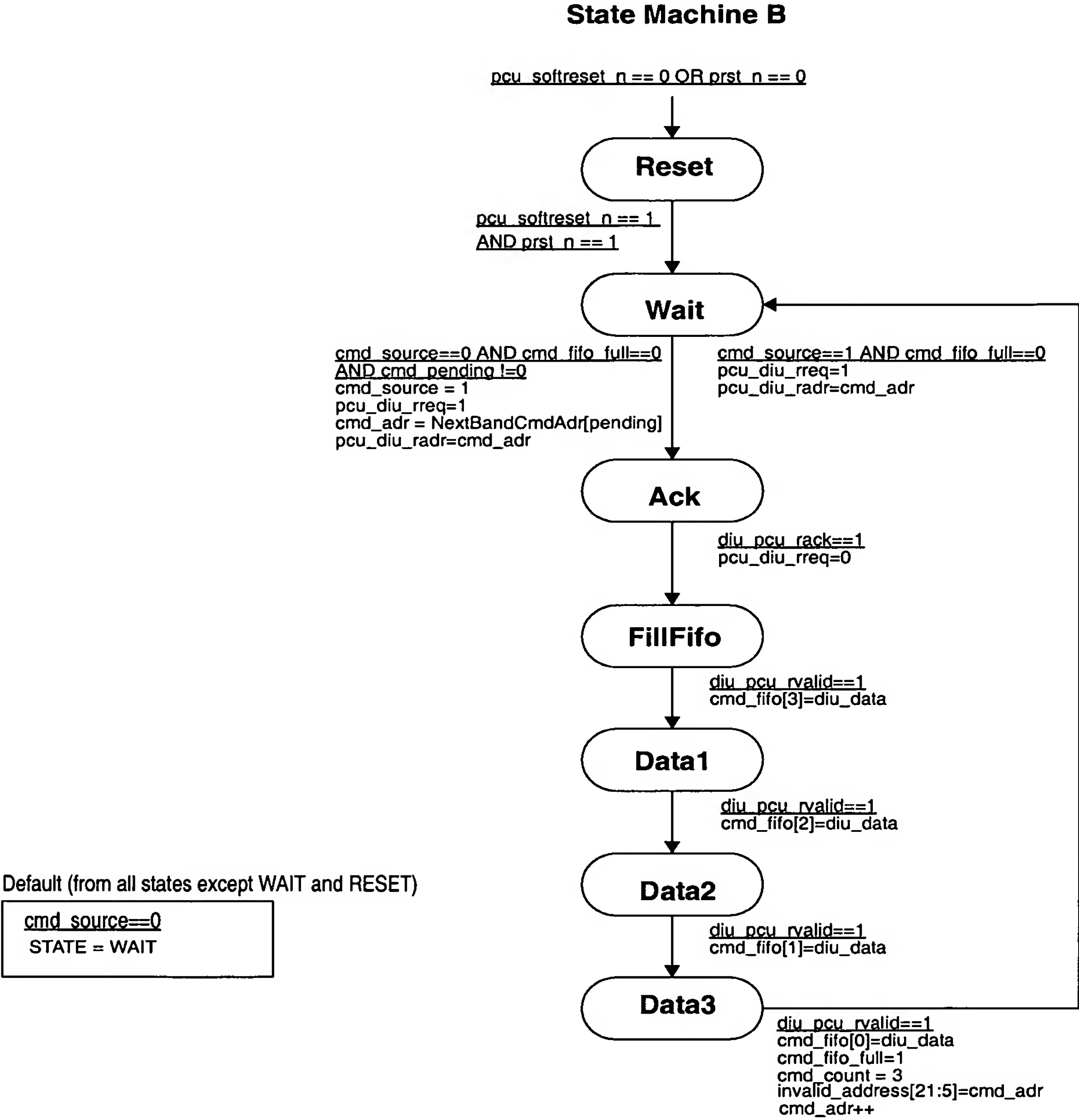


FIG. 135

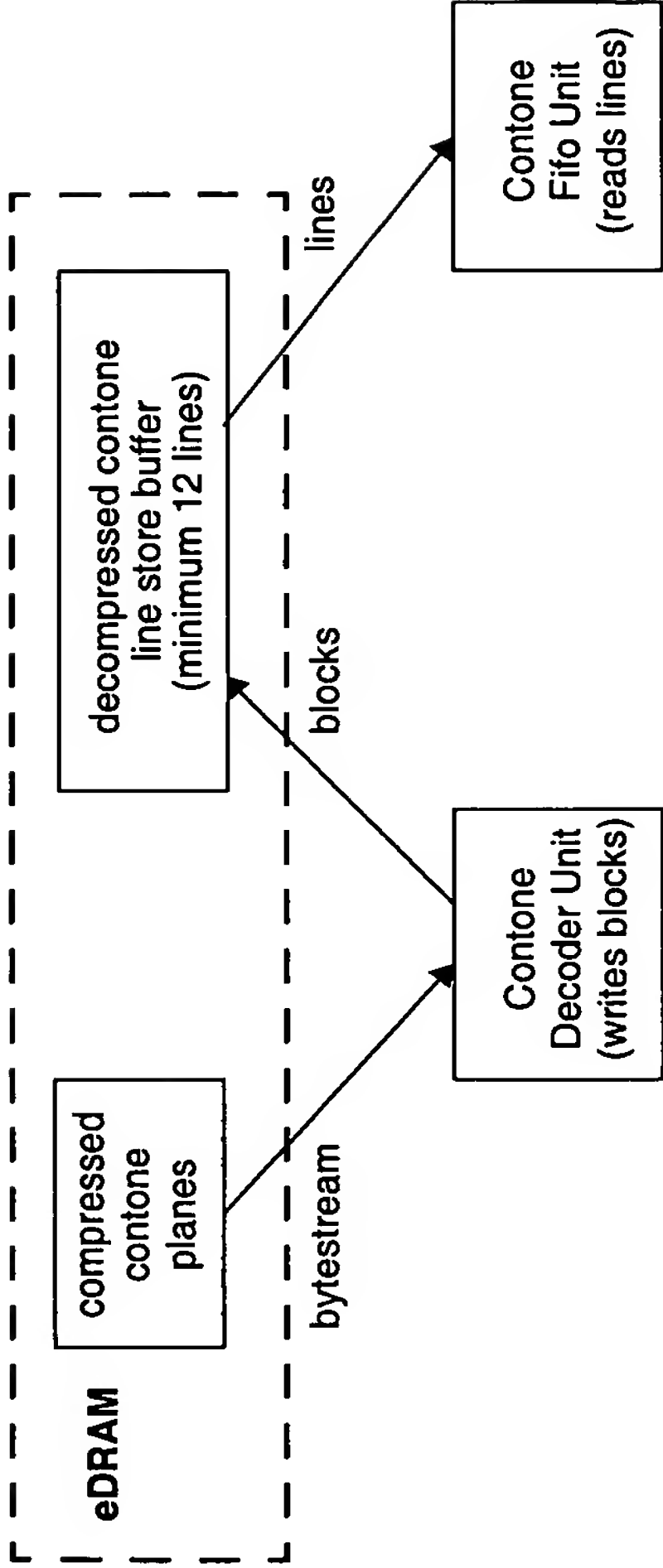


FIG. 136

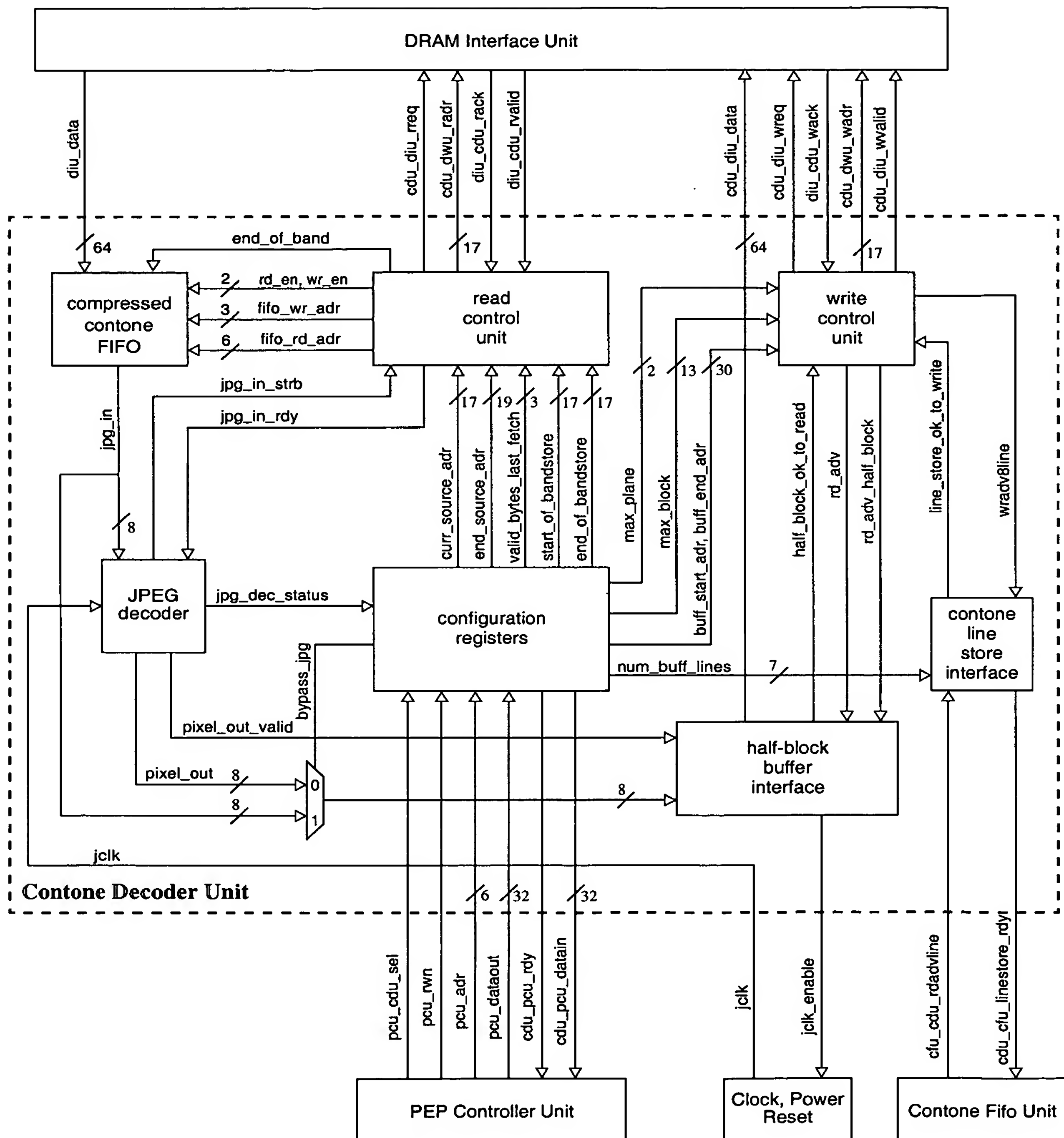


FIG. 137

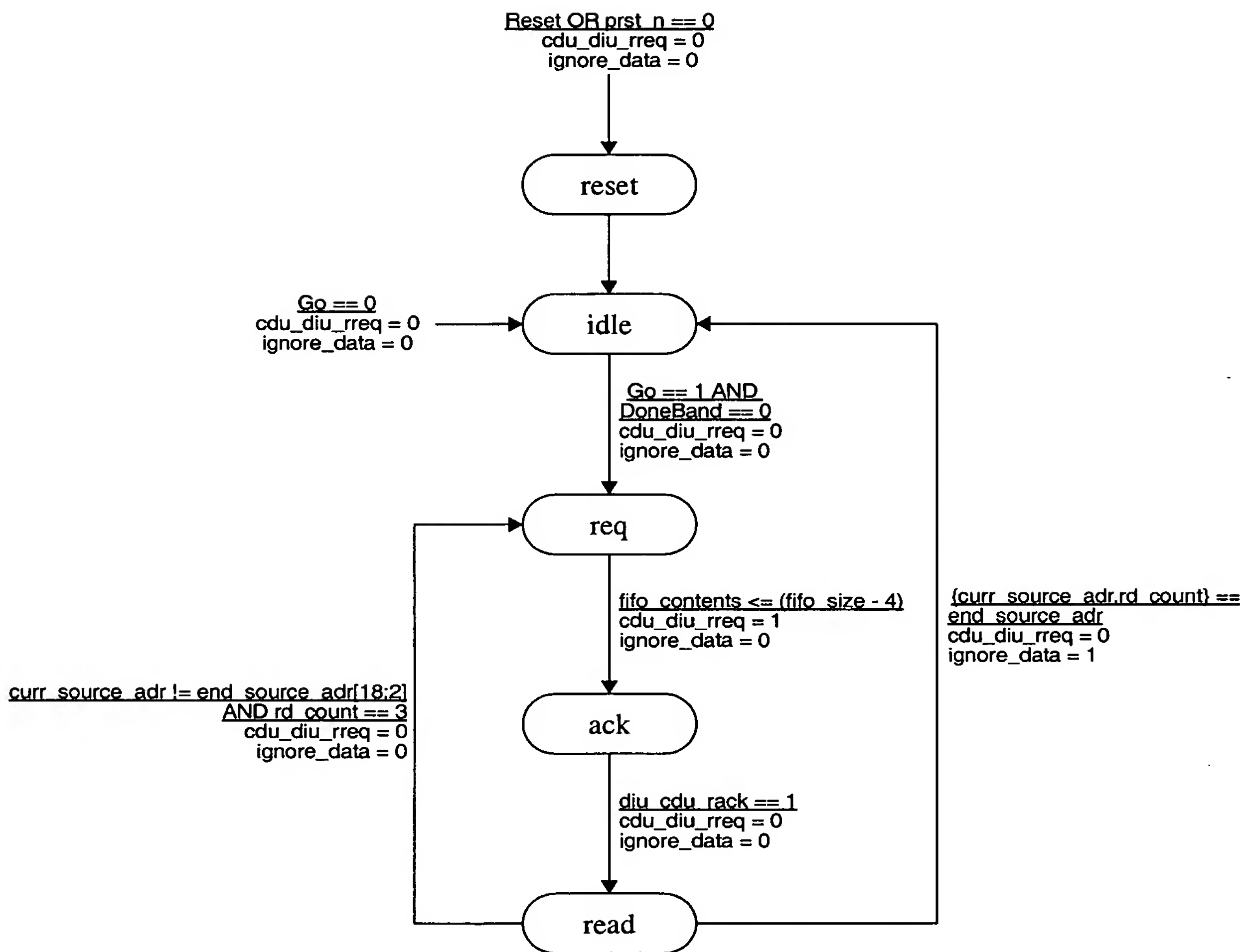


FIG. 138

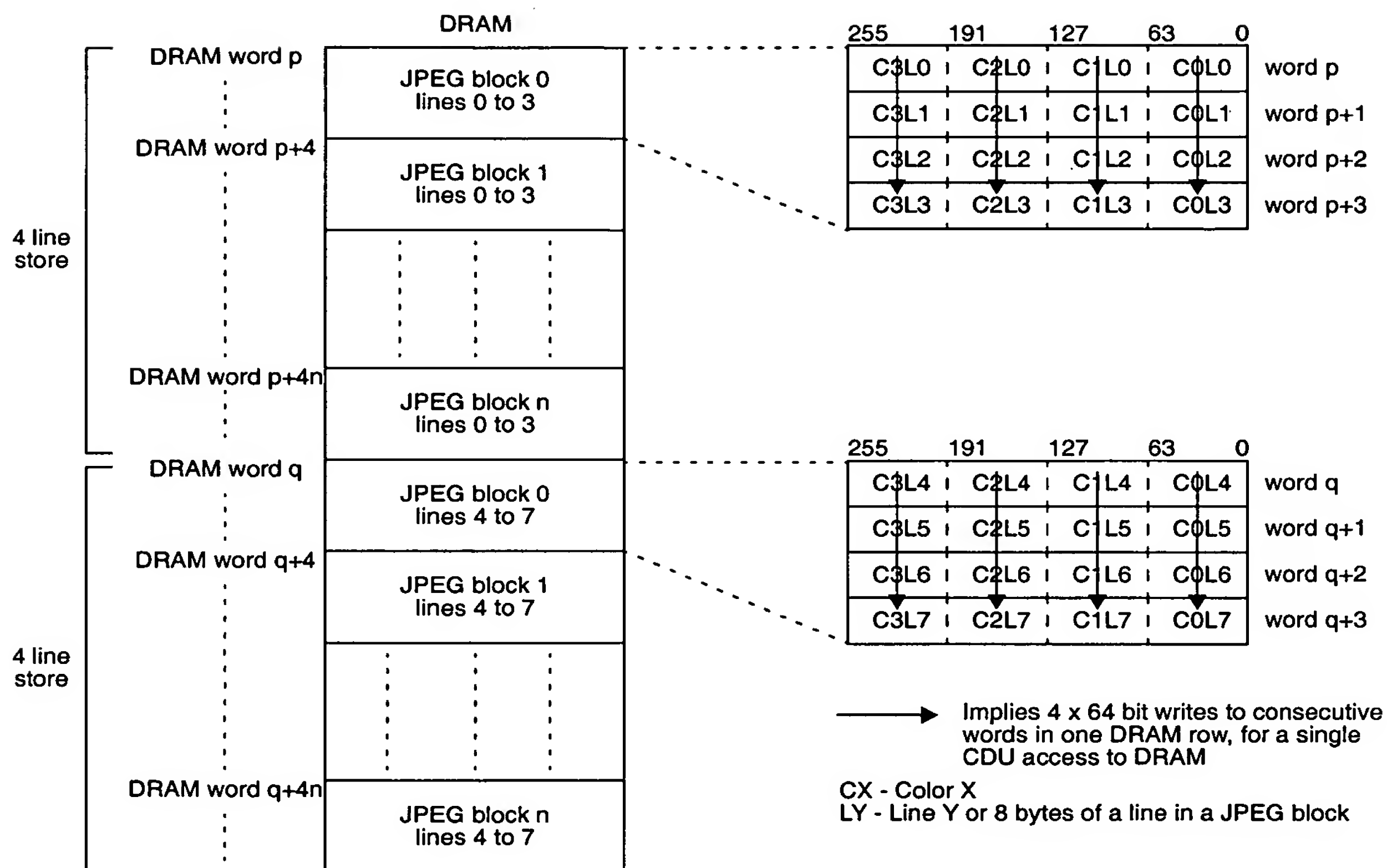


FIG. 139

121/331

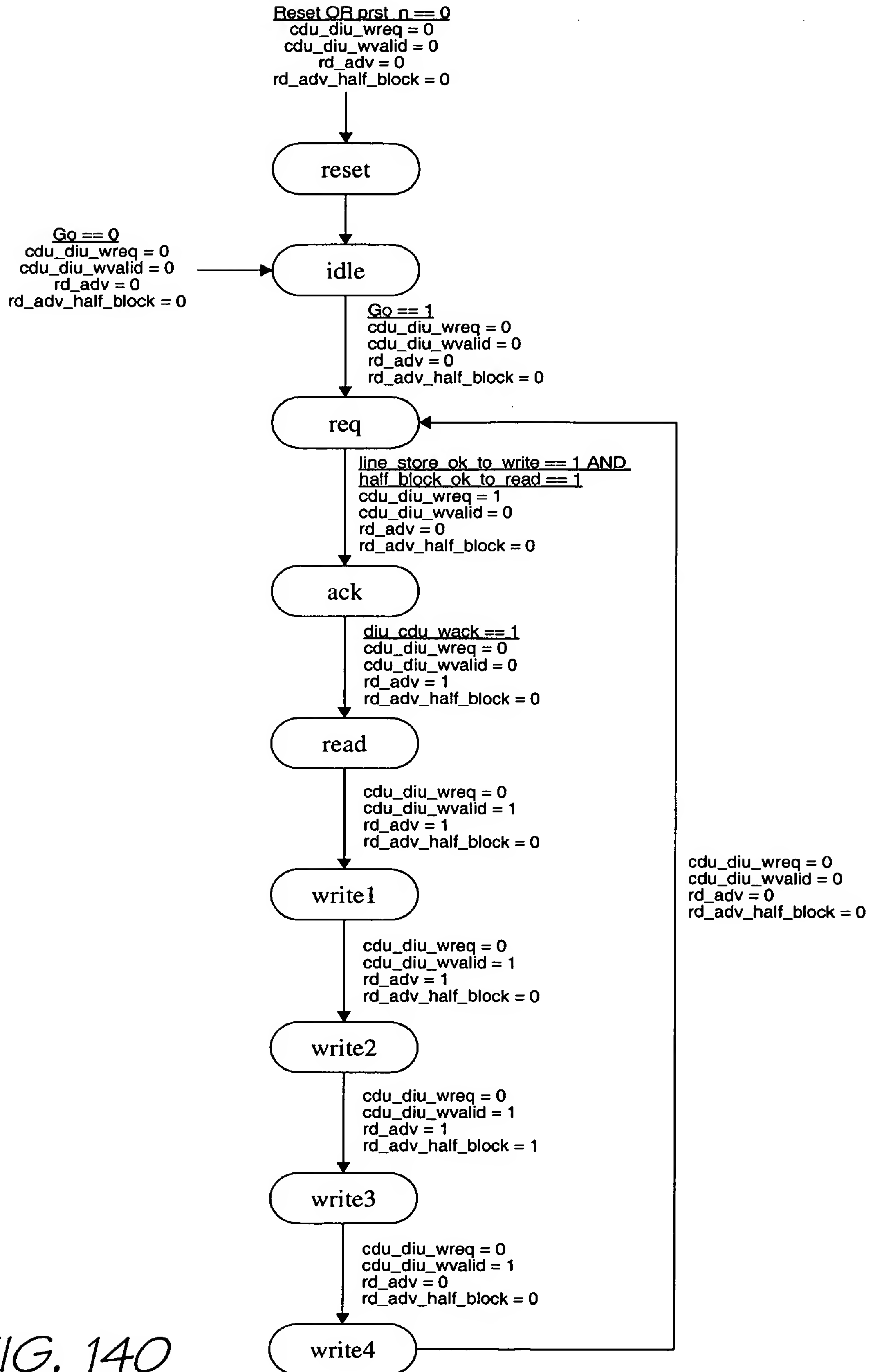


FIG. 140

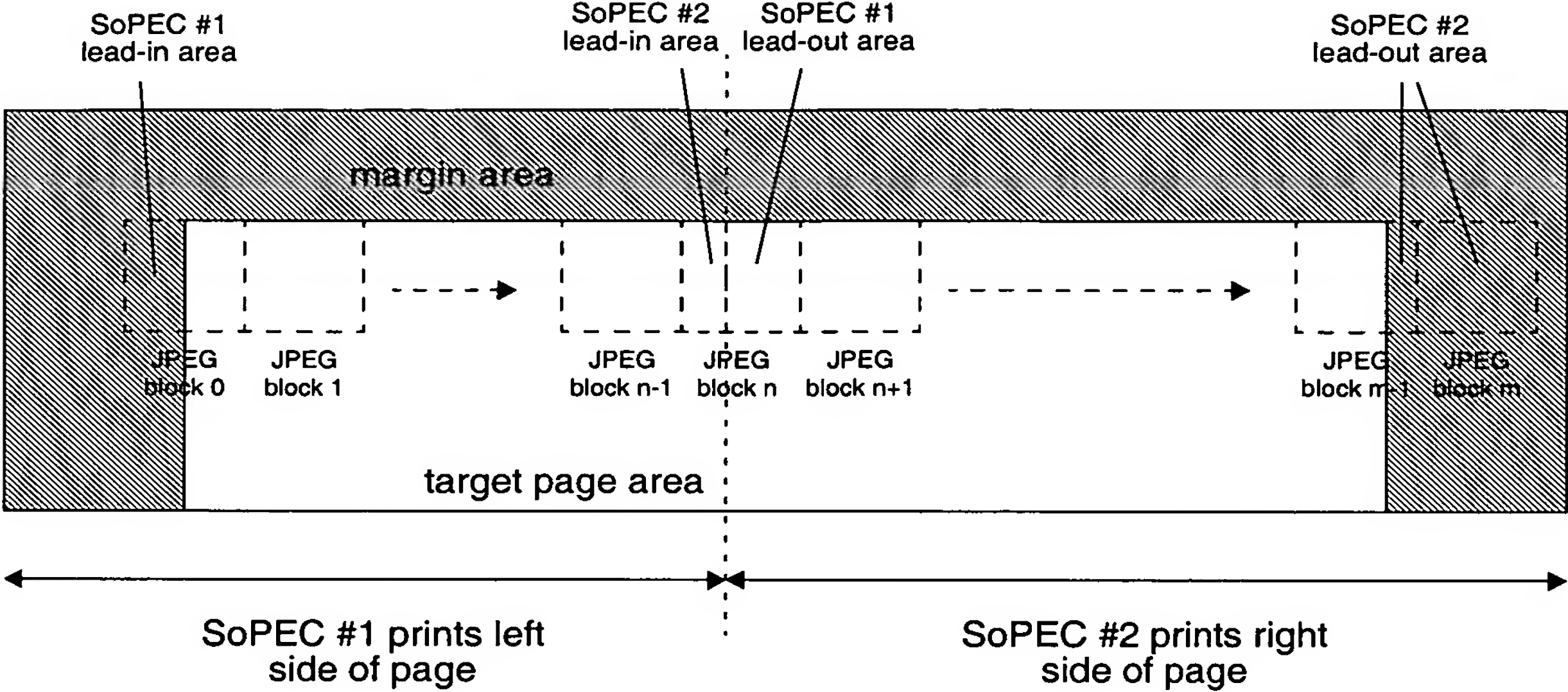


FIG. 141

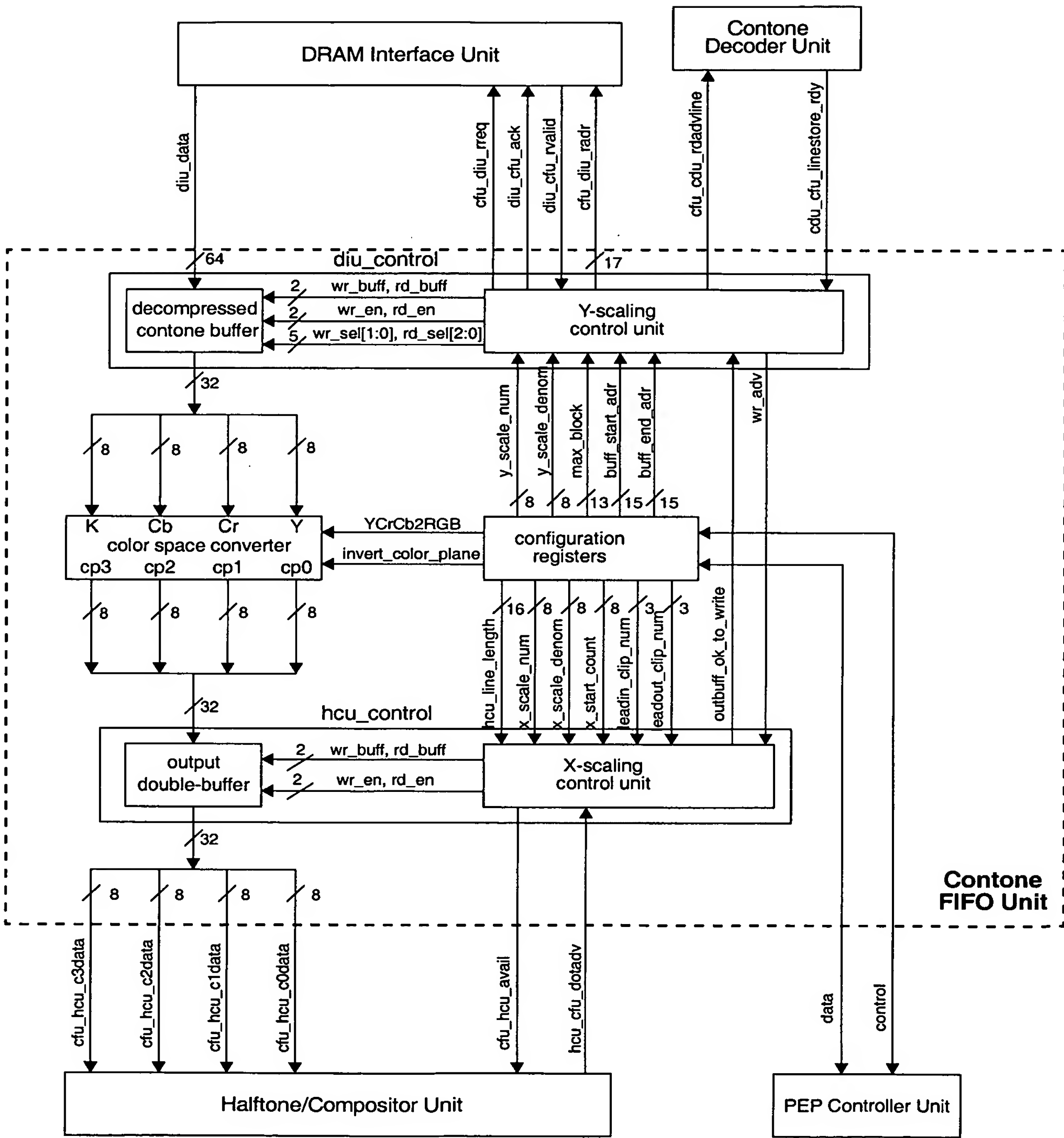


FIG. 142

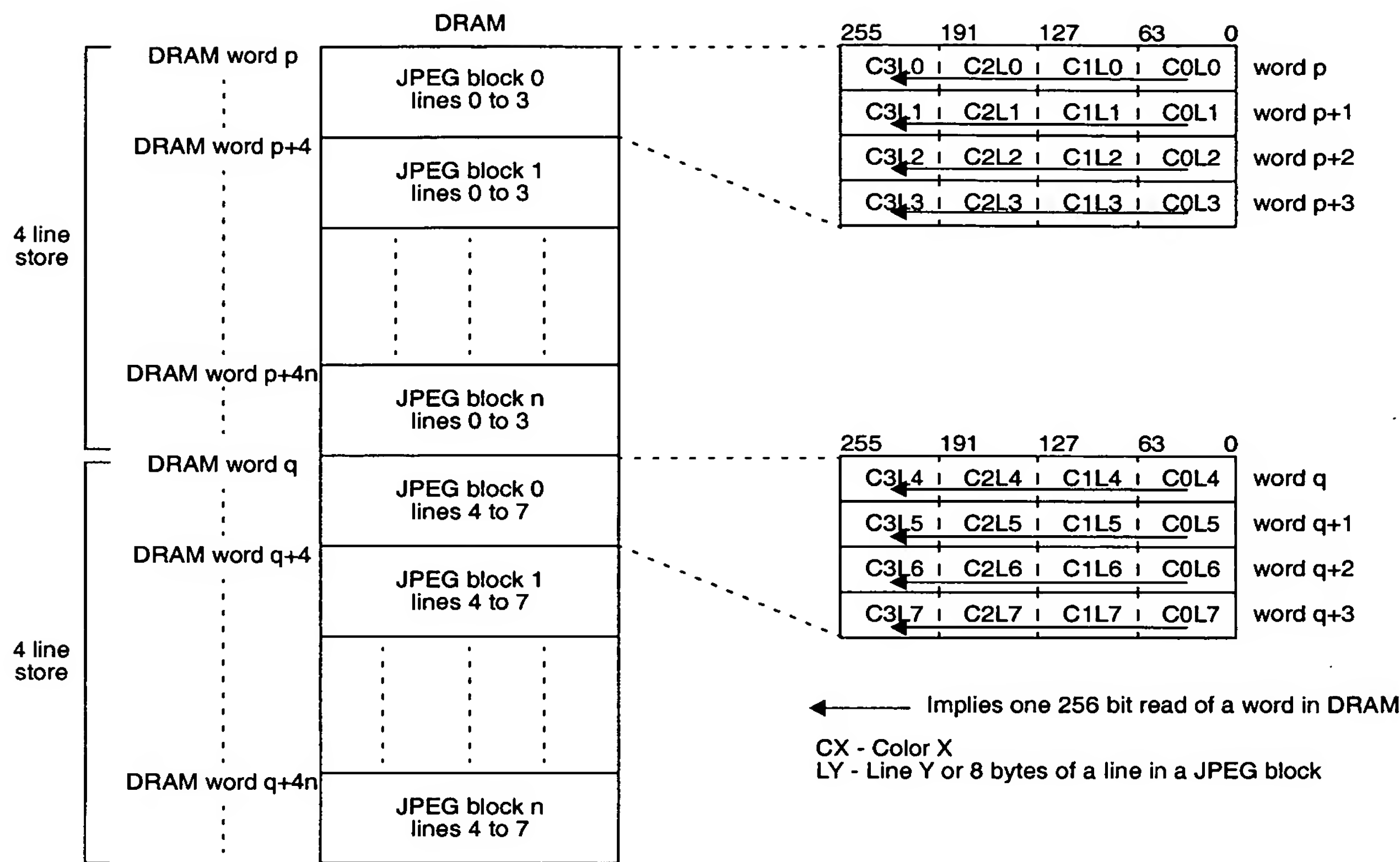


FIG. 143

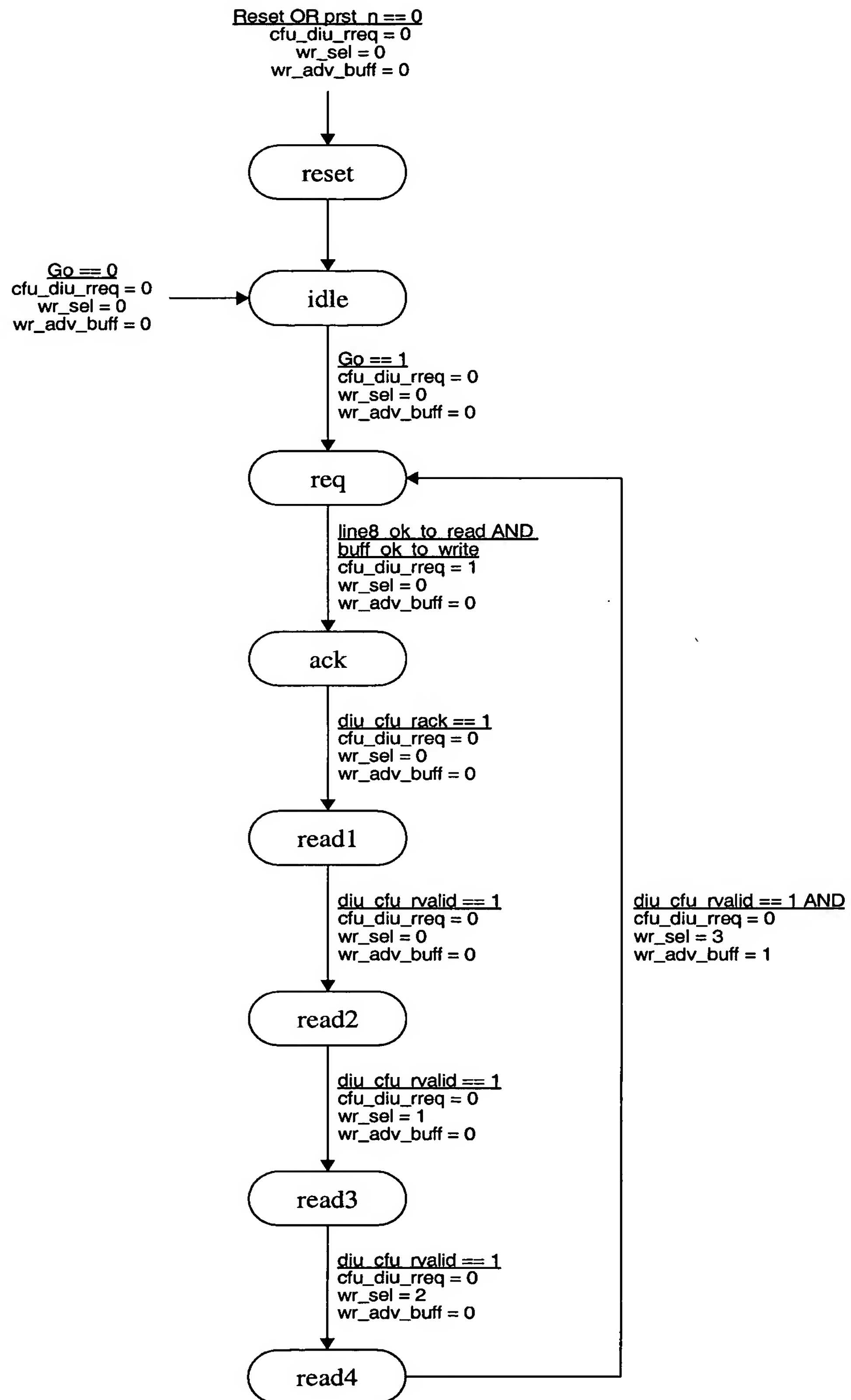


FIG. 144

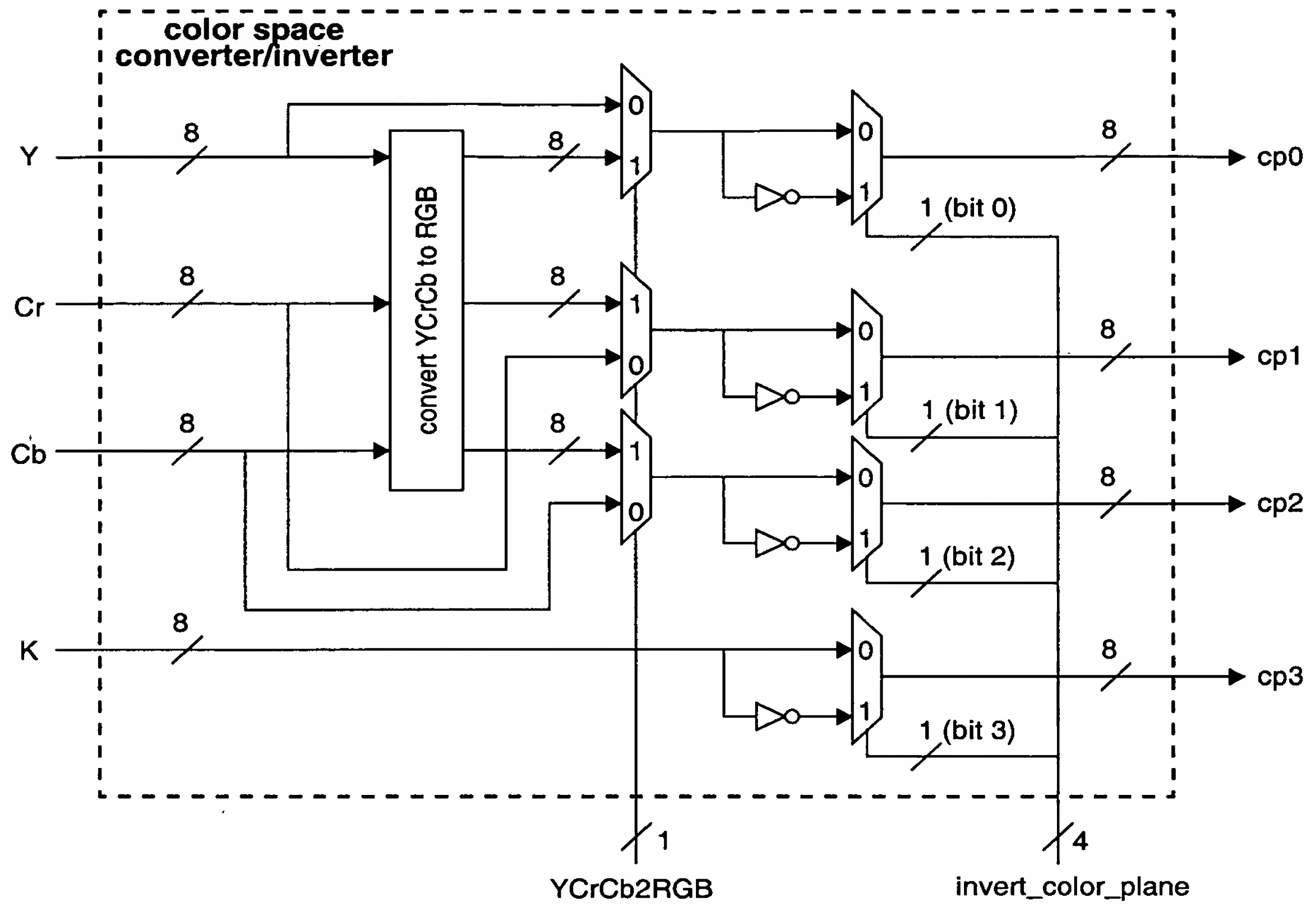


FIG. 145

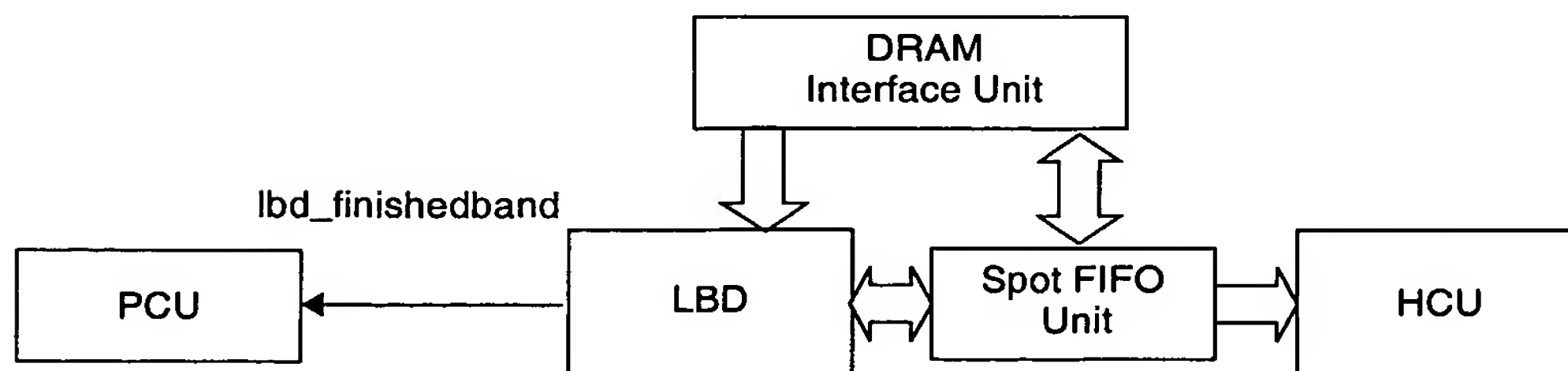


FIG. 146

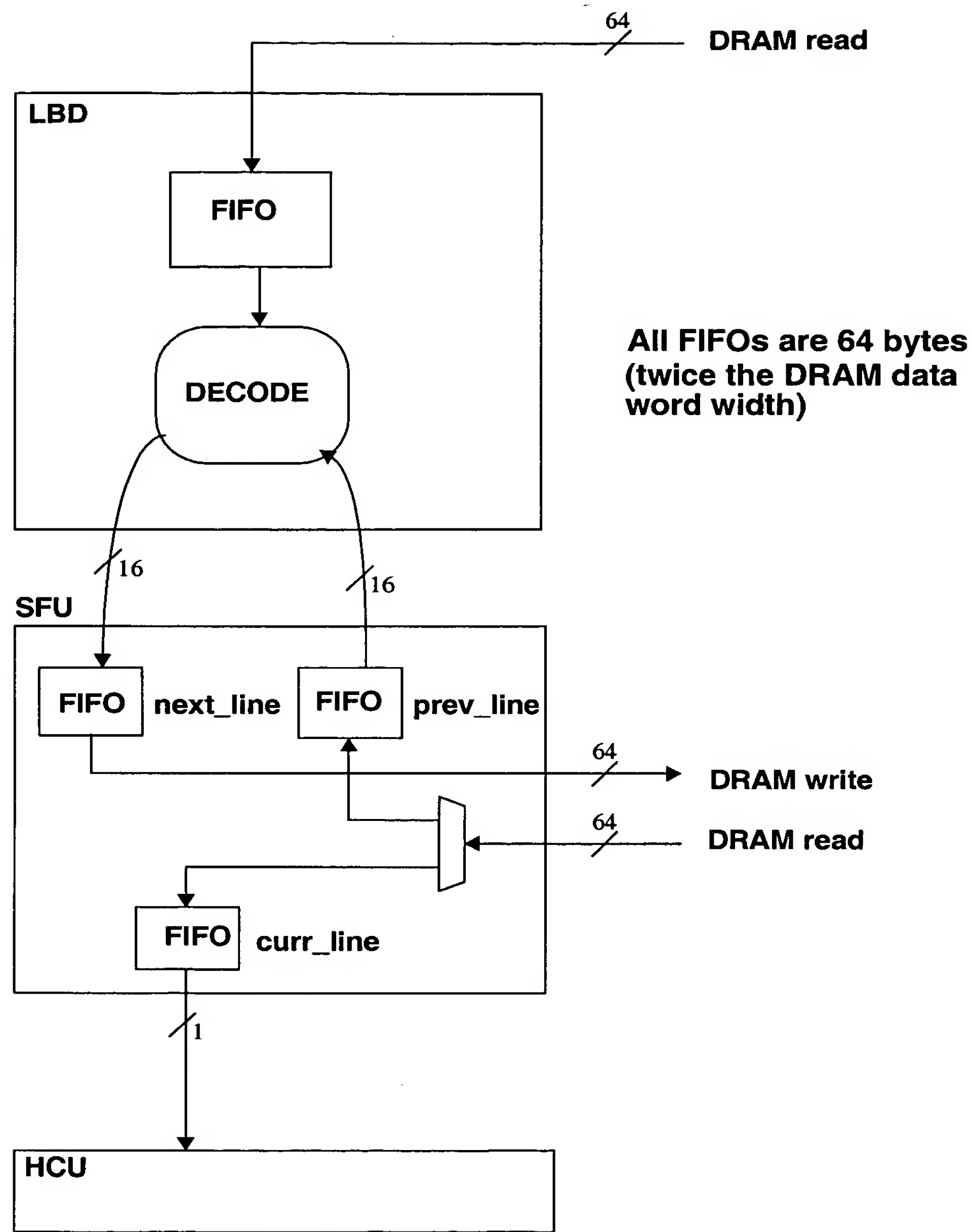


FIG. 147

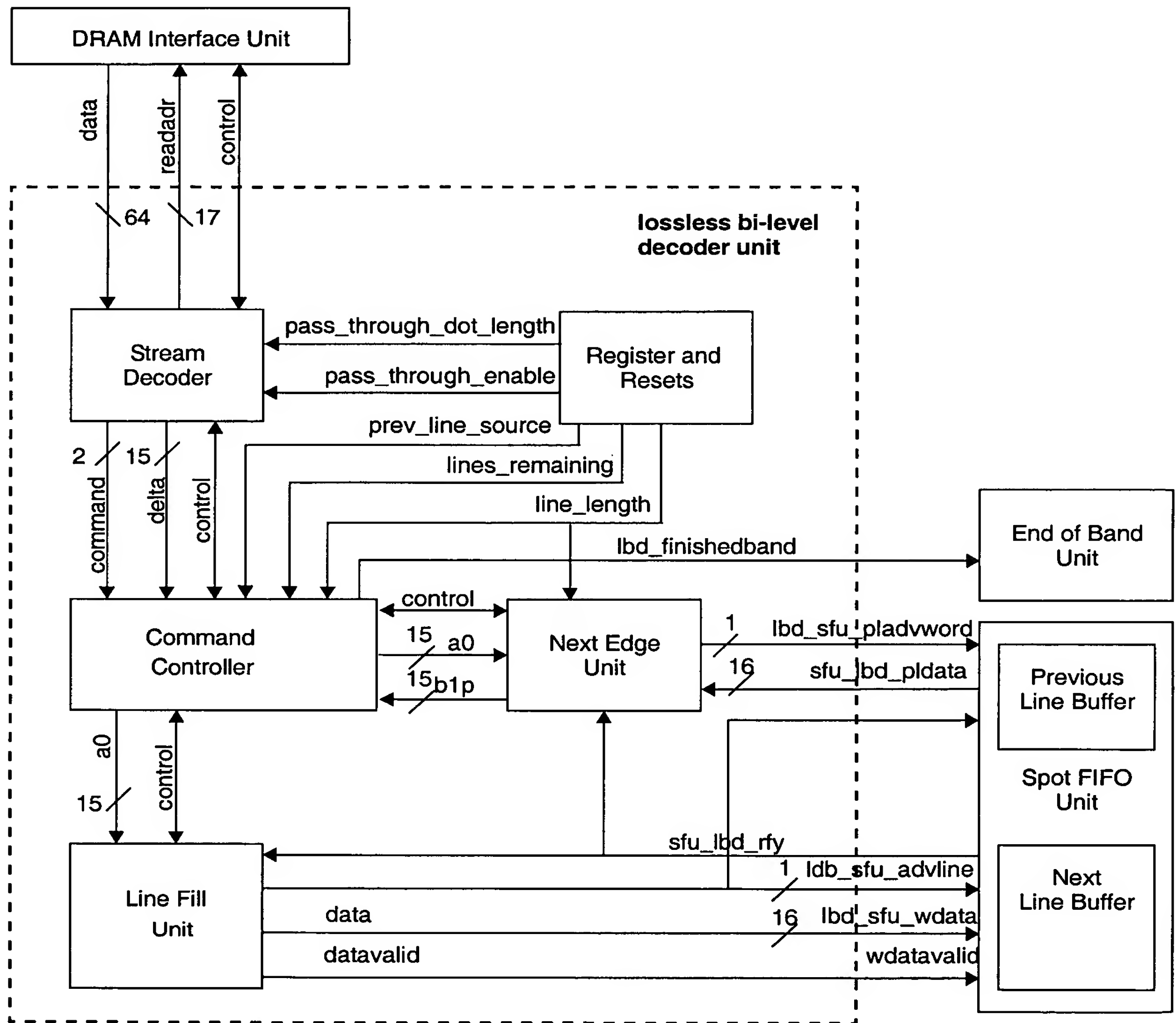


FIG. 148

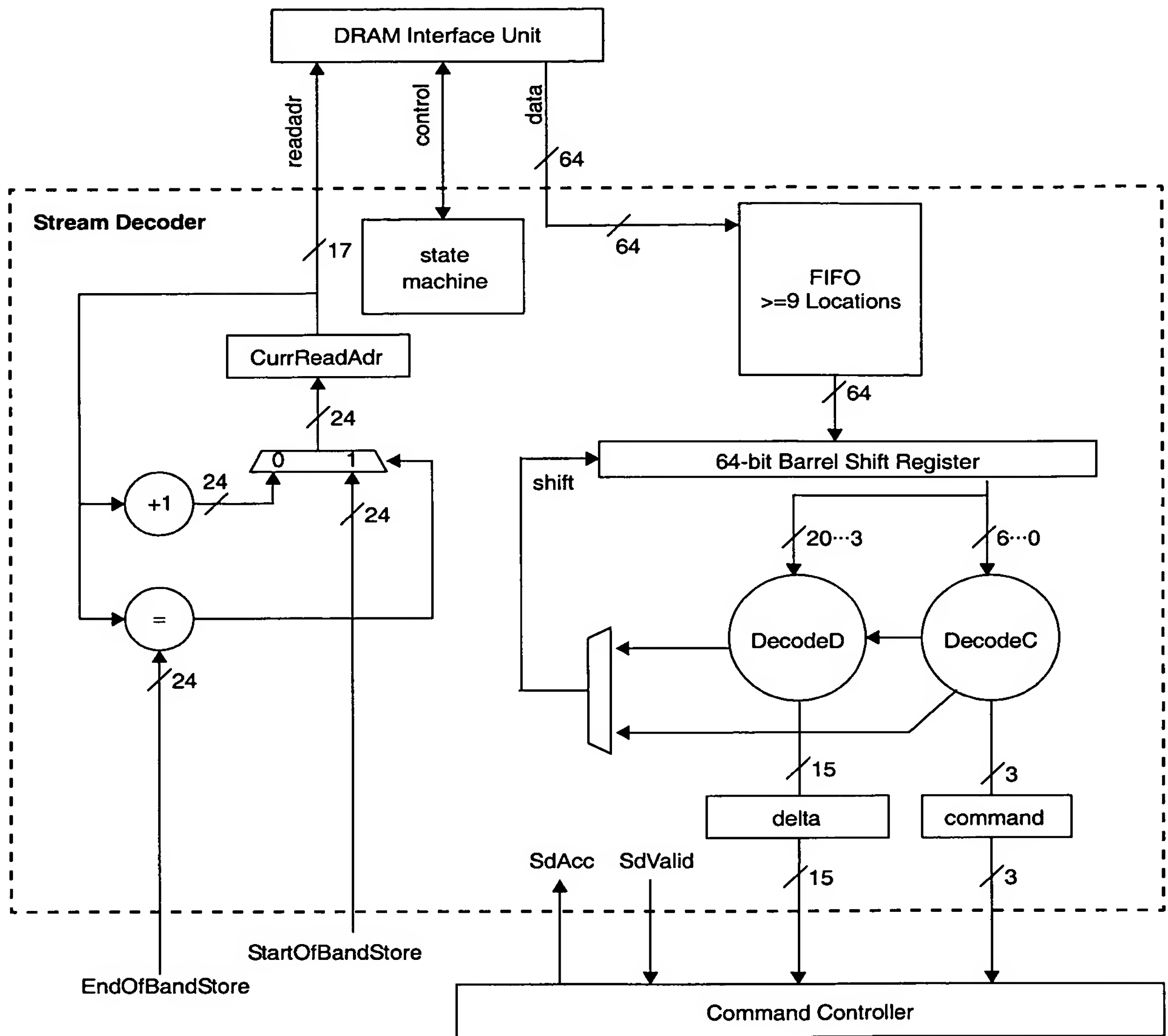


FIG. 149

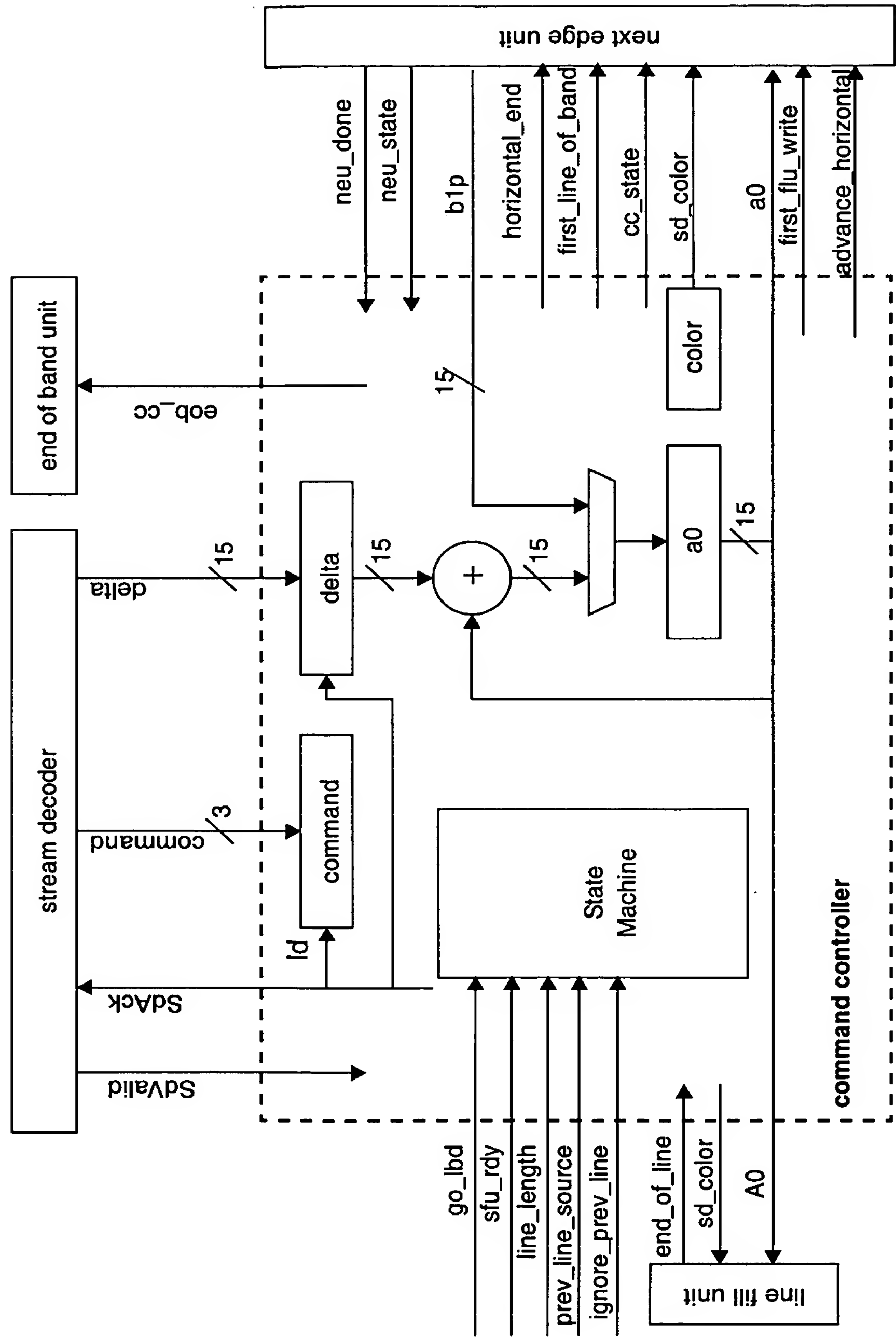


FIG. 150

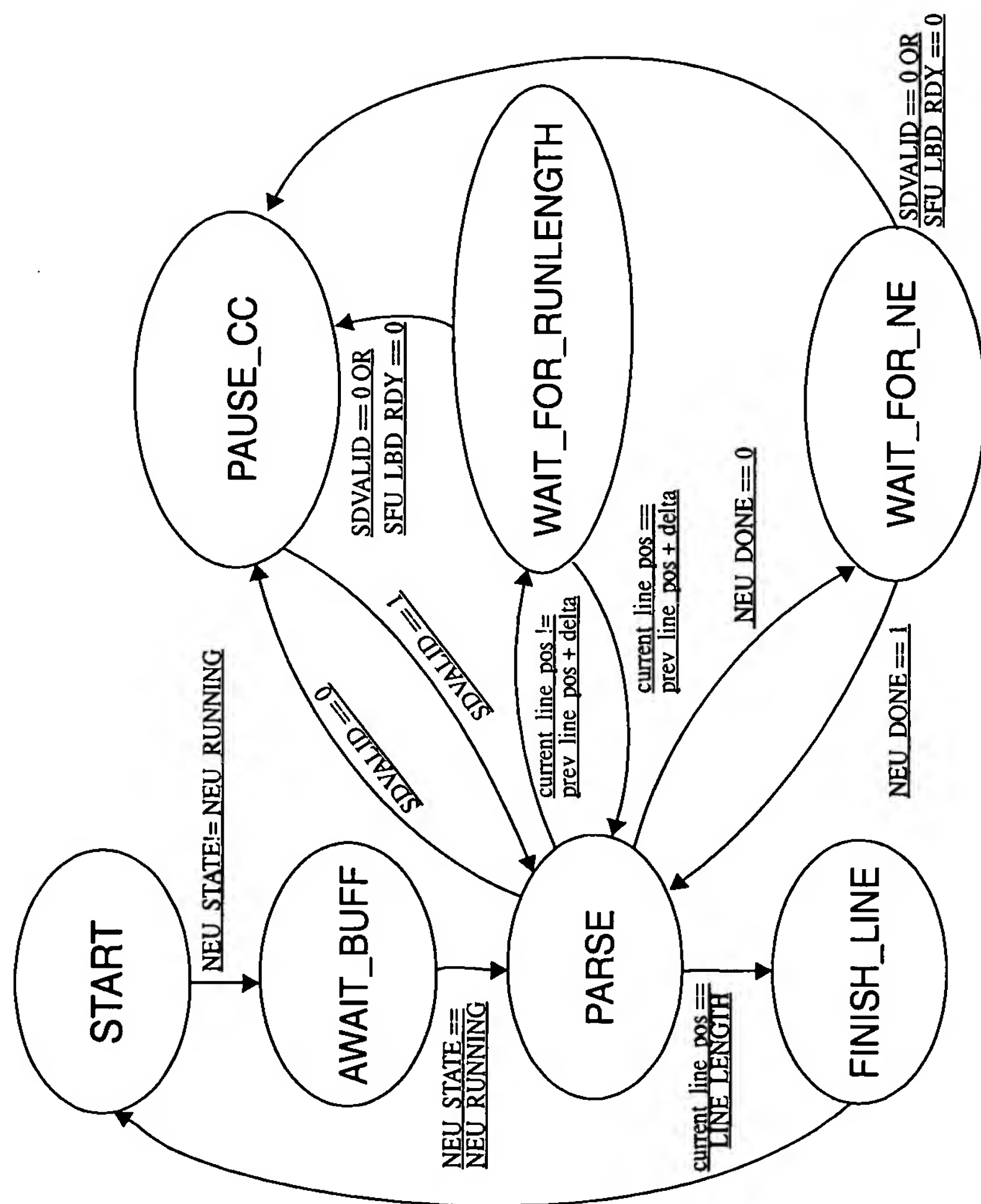


FIG. 151

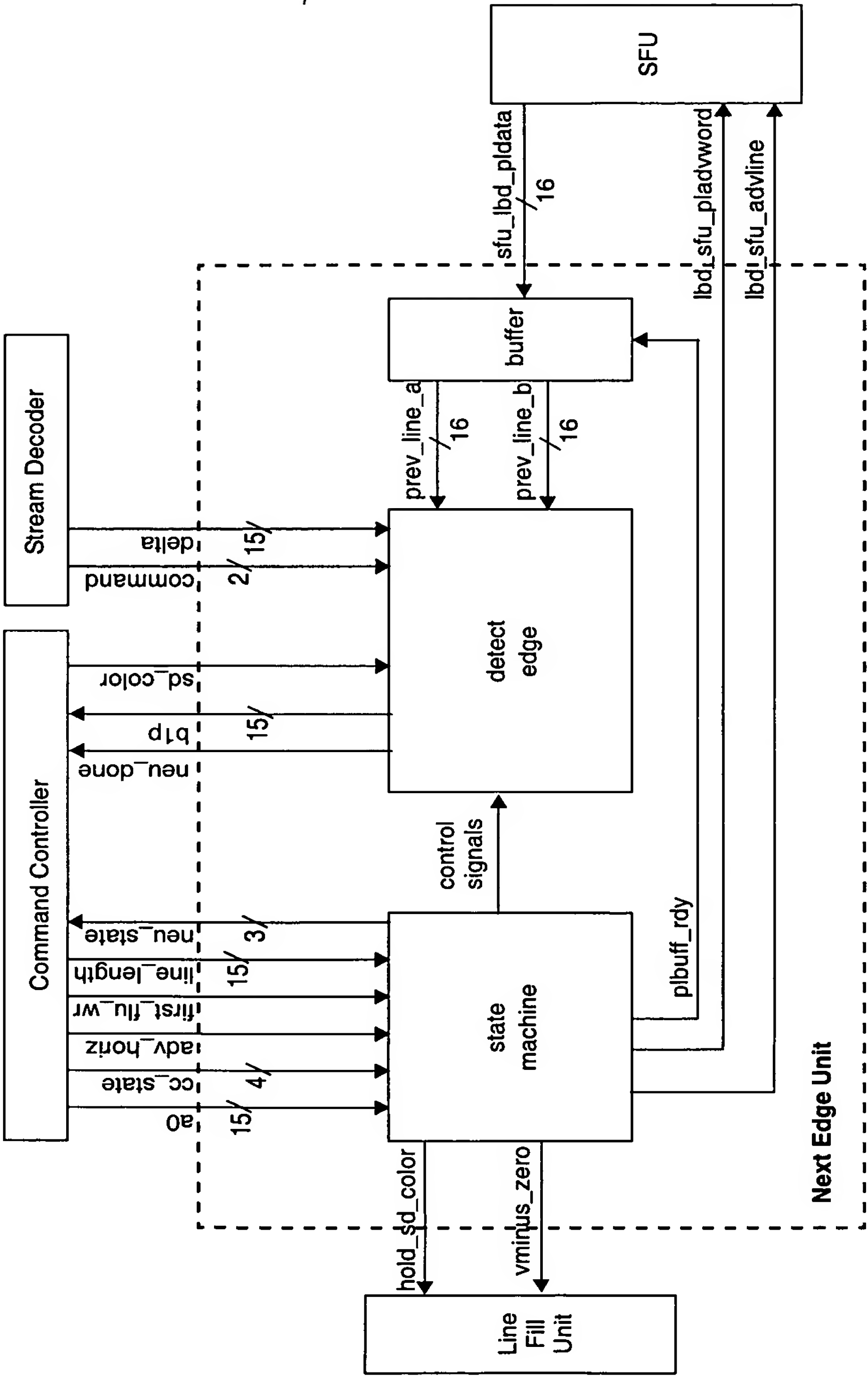
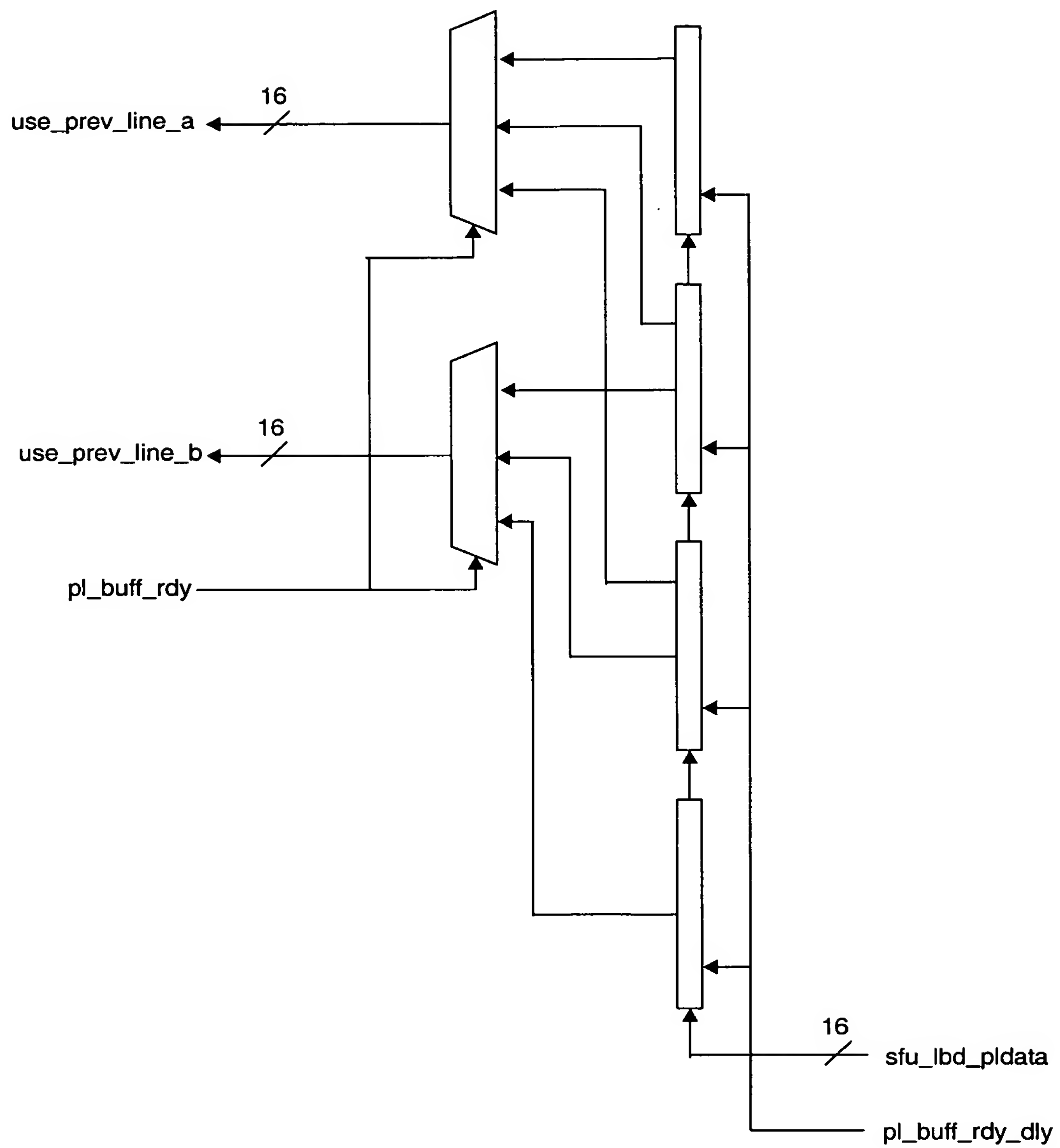


FIG. 152

*FIG. 153*

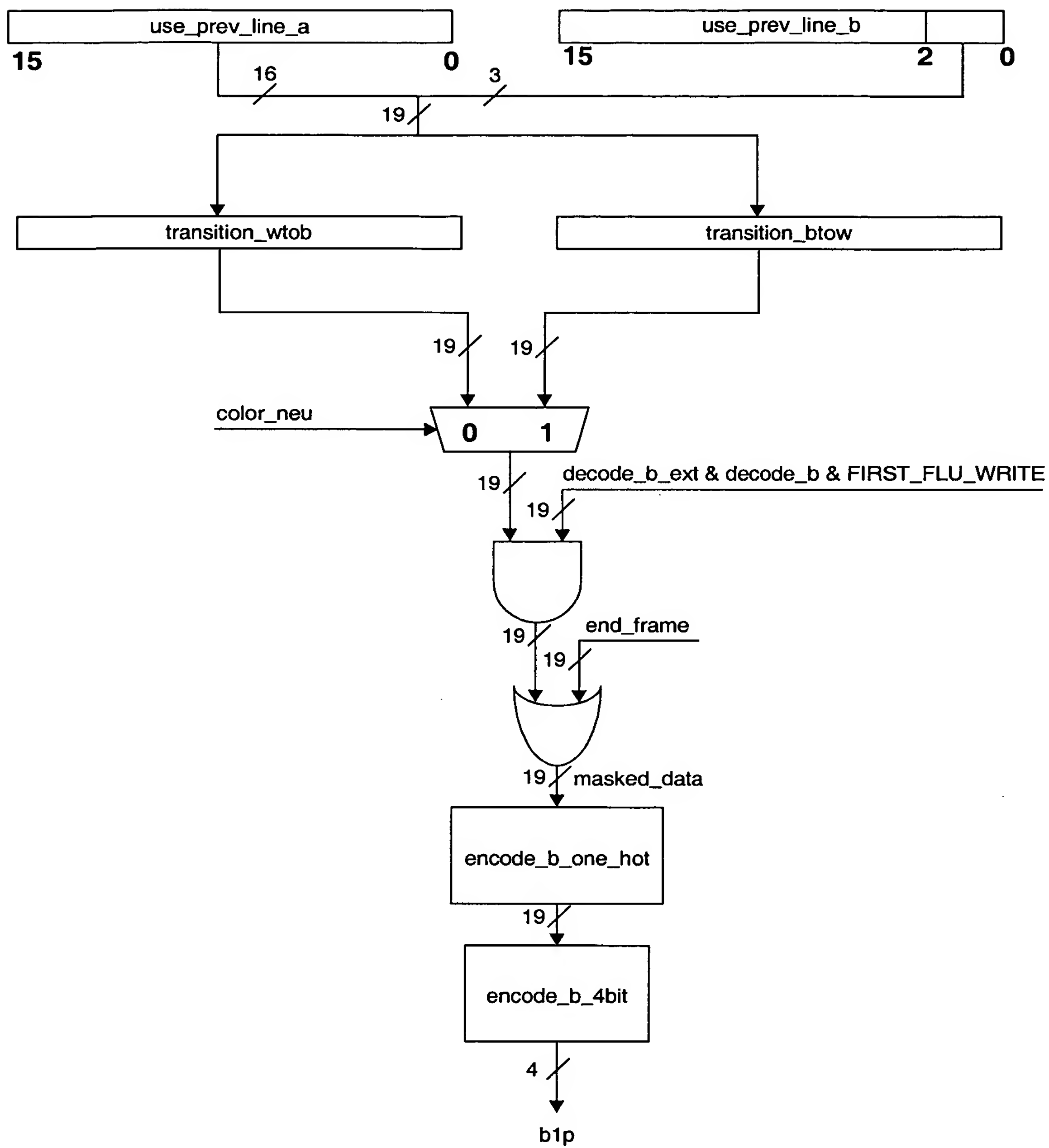
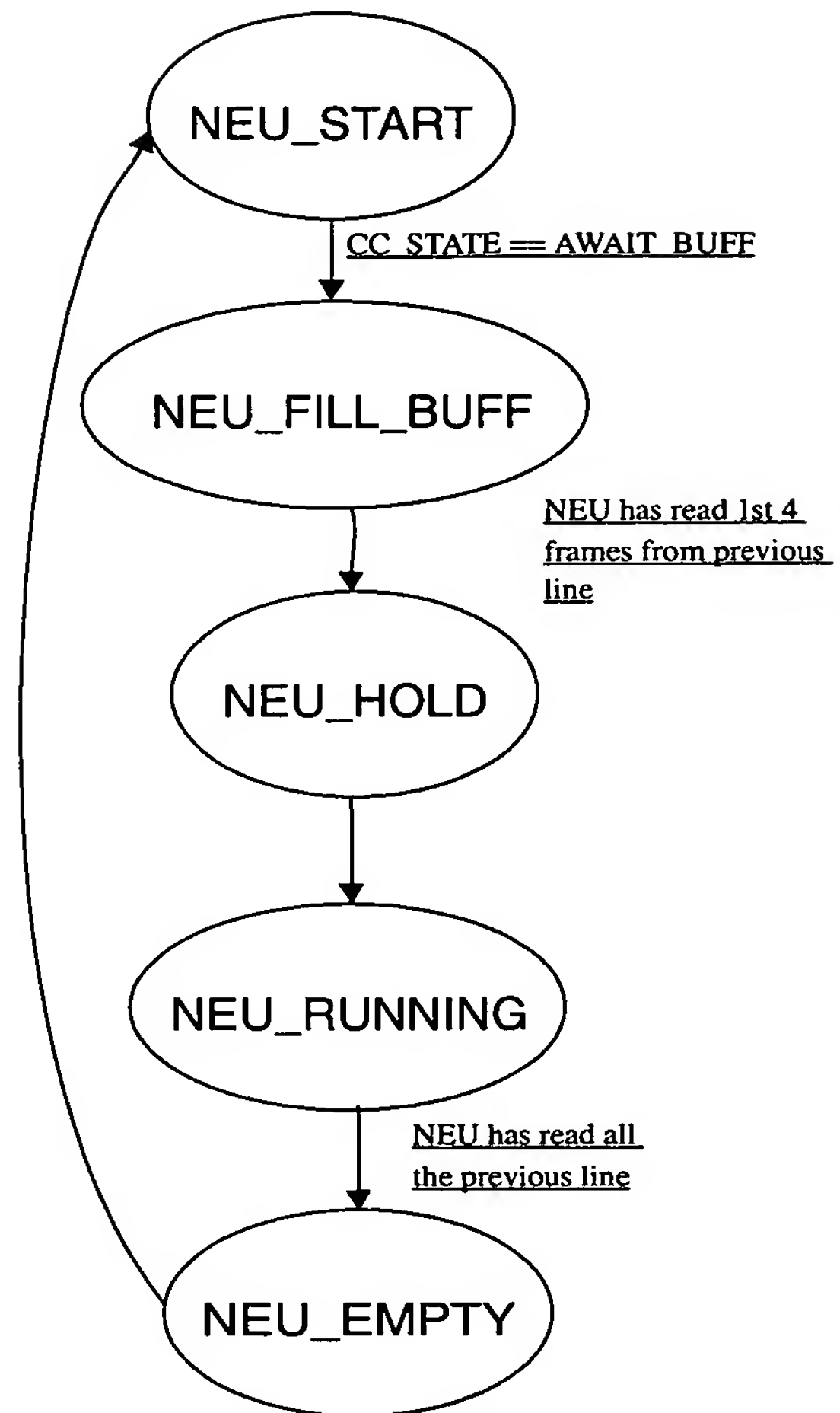


FIG. 154

*FIG. 155*

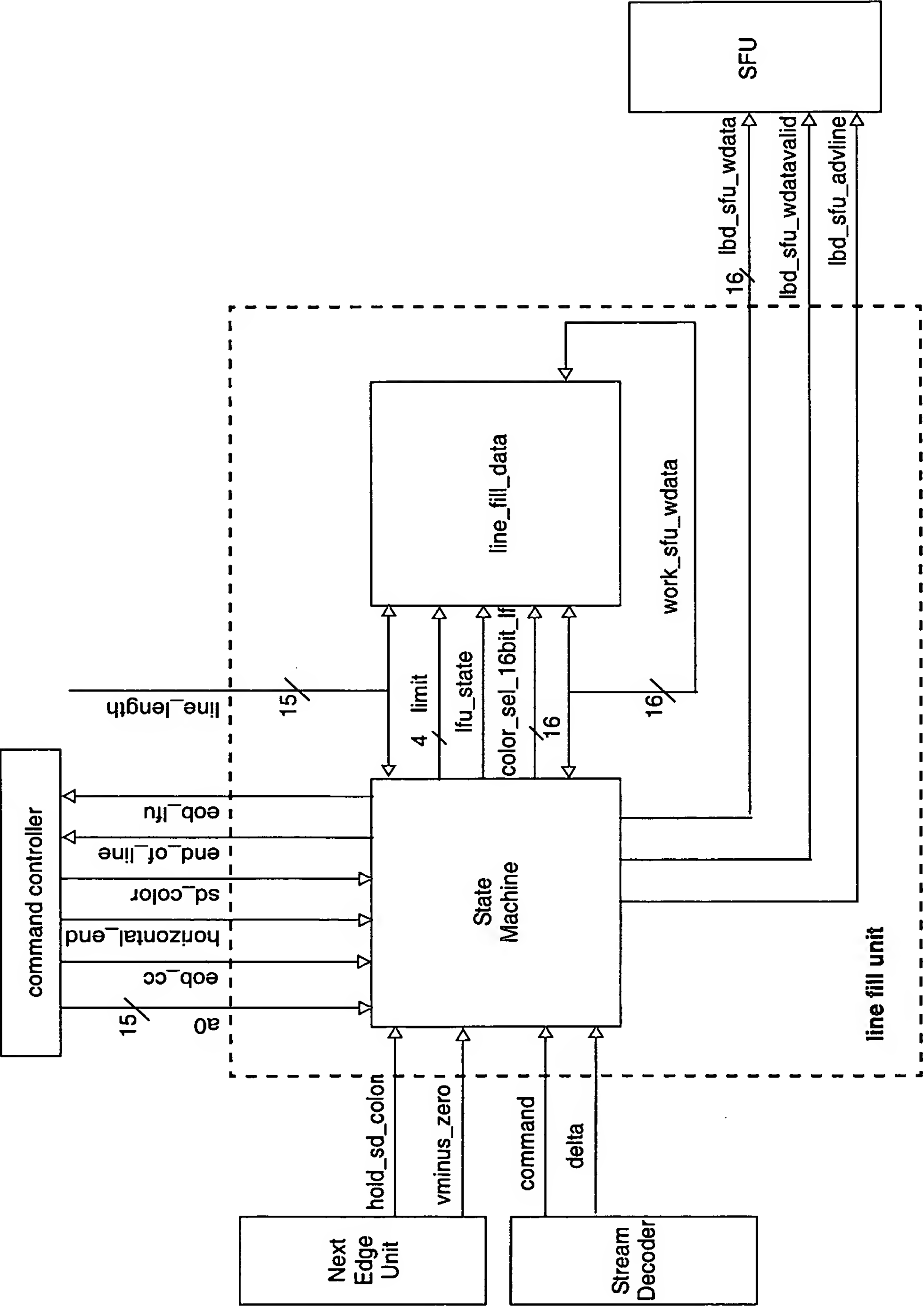


FIG. 156

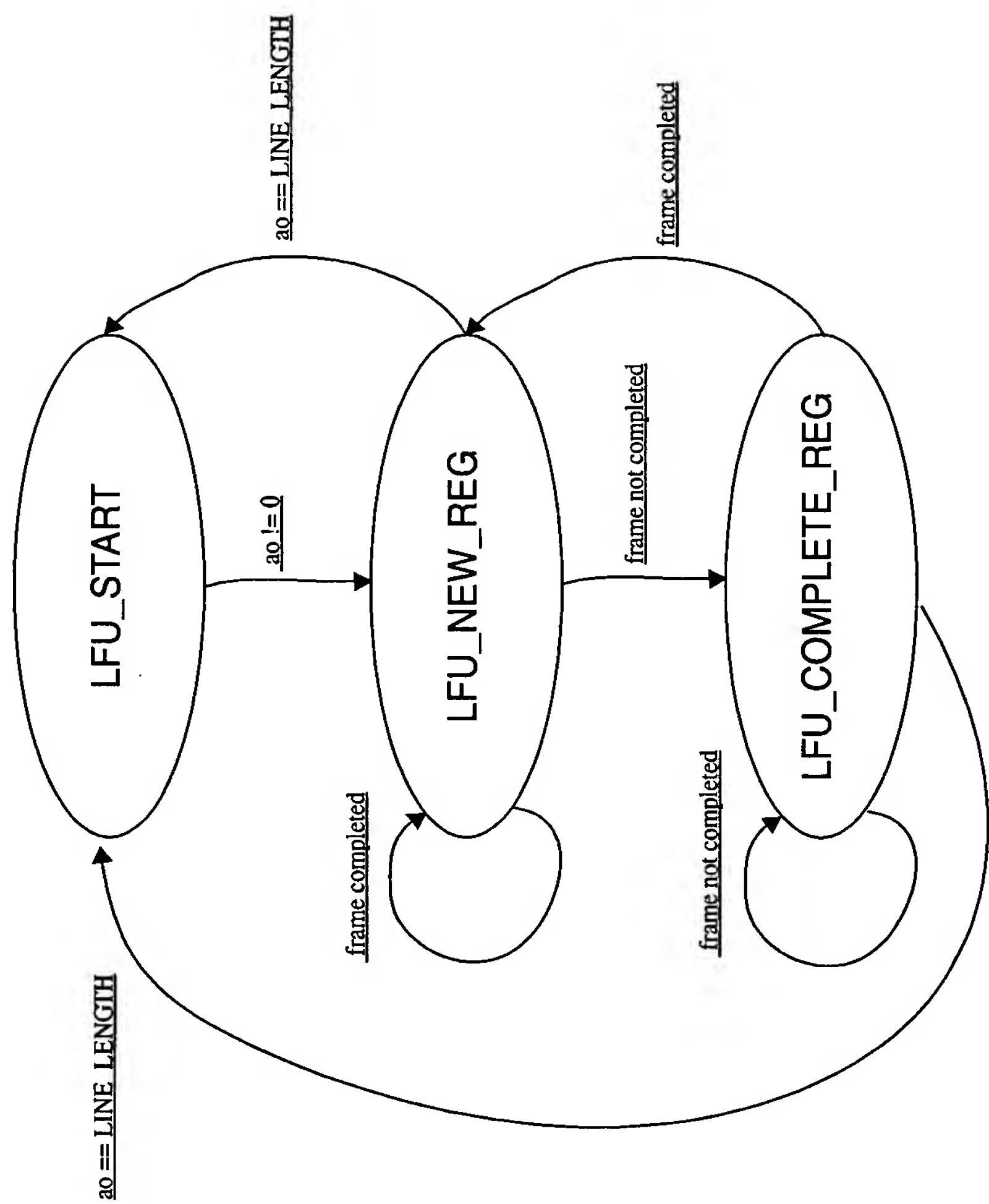


FIG. 157

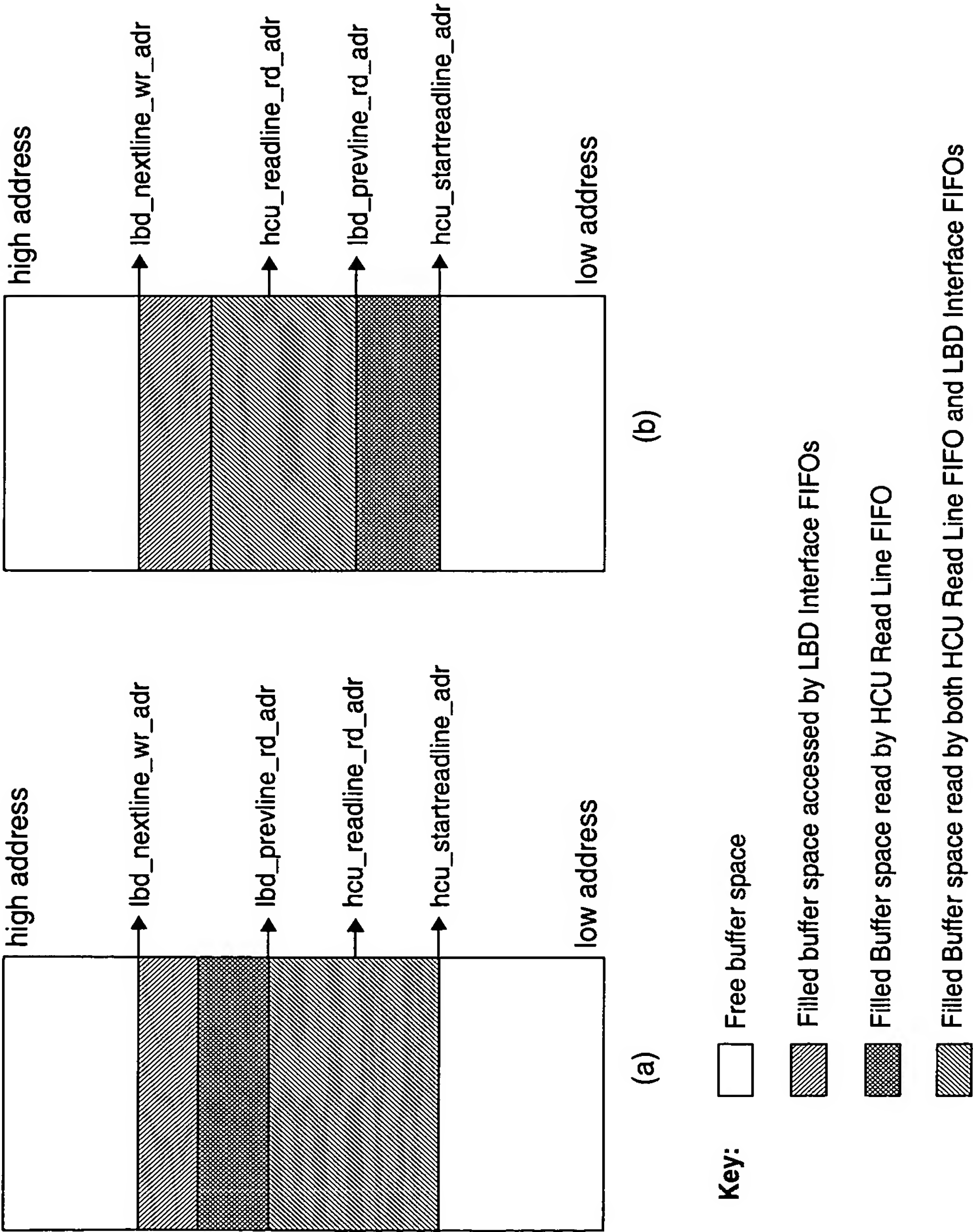


FIG. 158

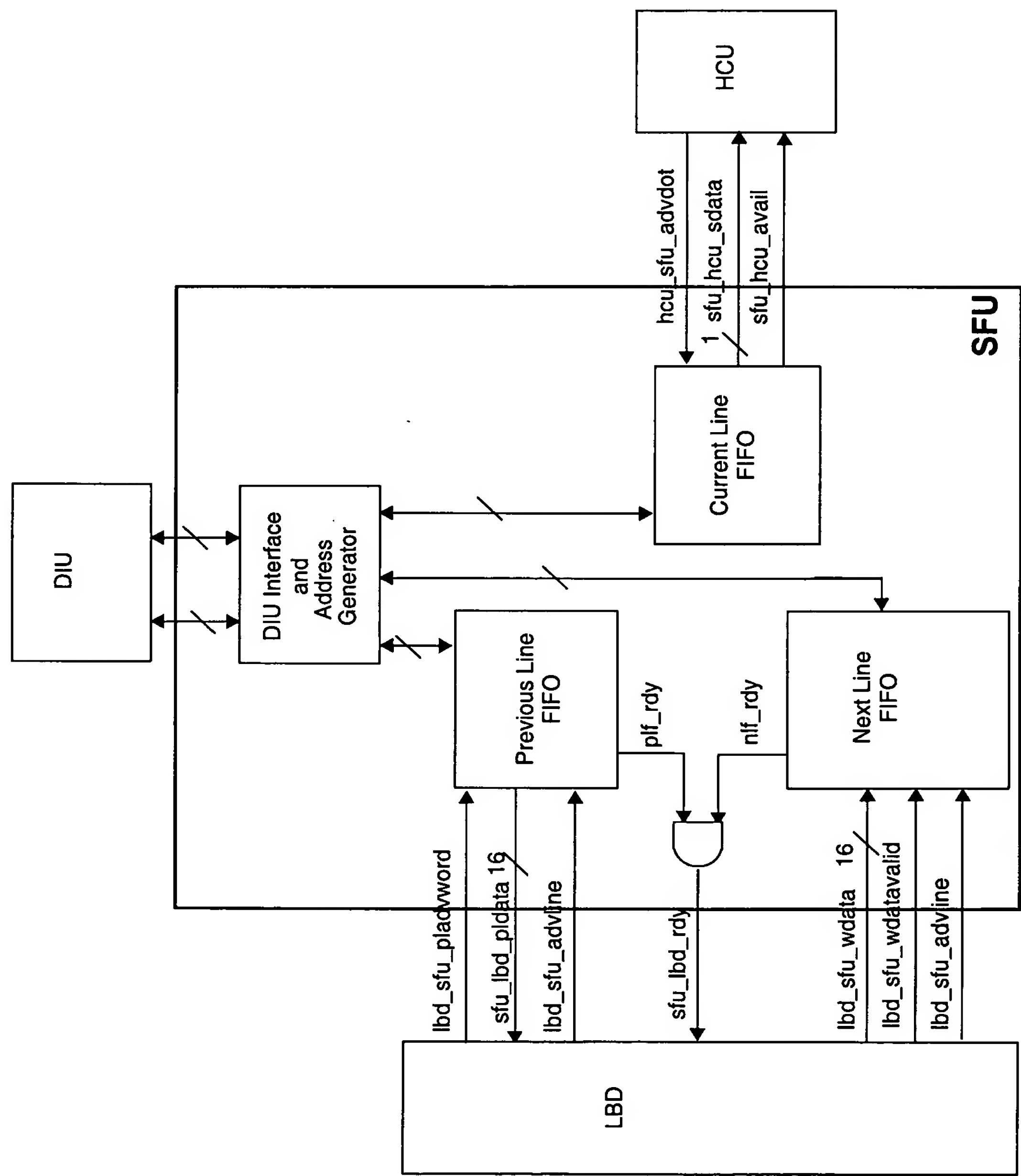


FIG. 159

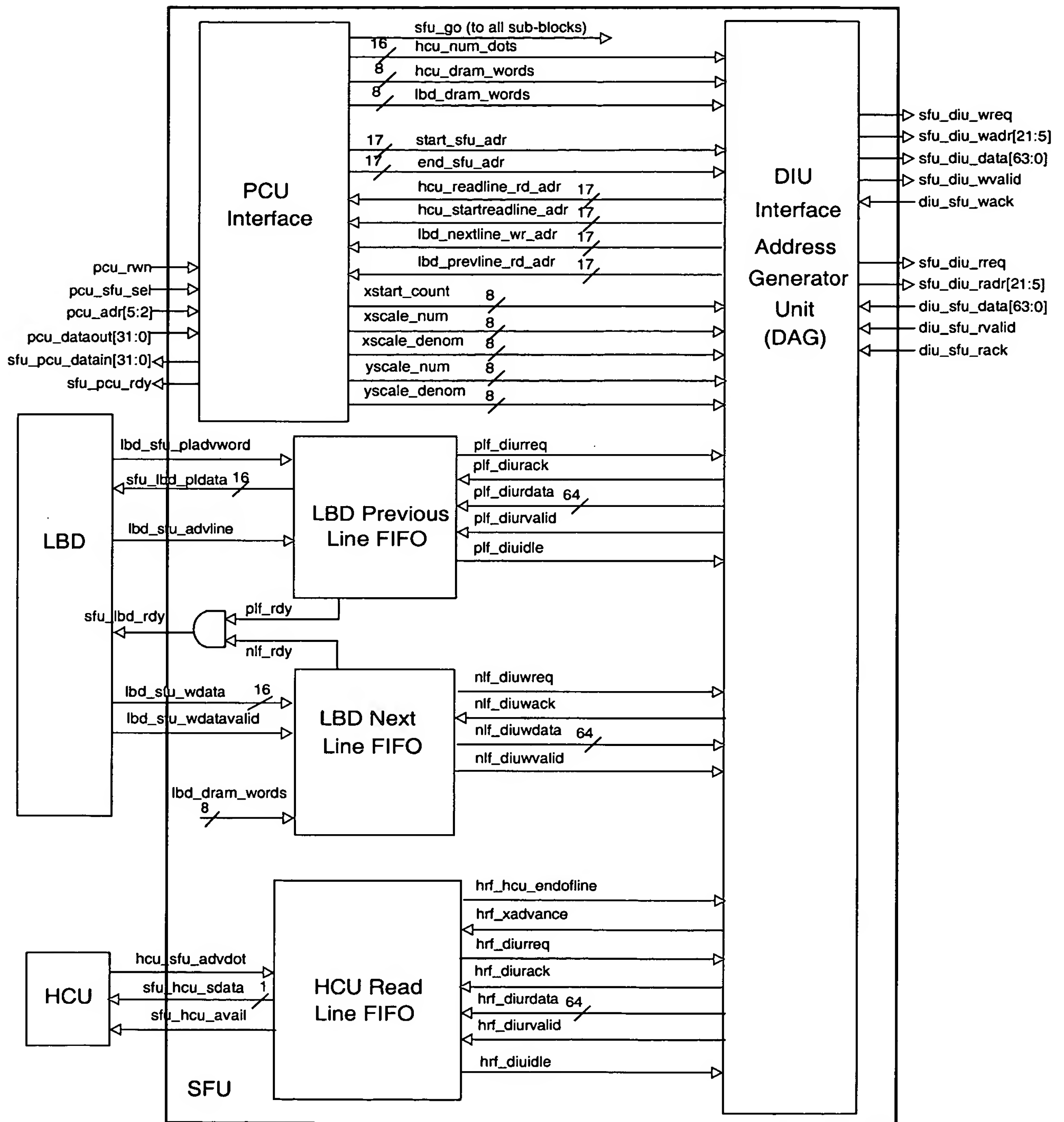


FIG. 160

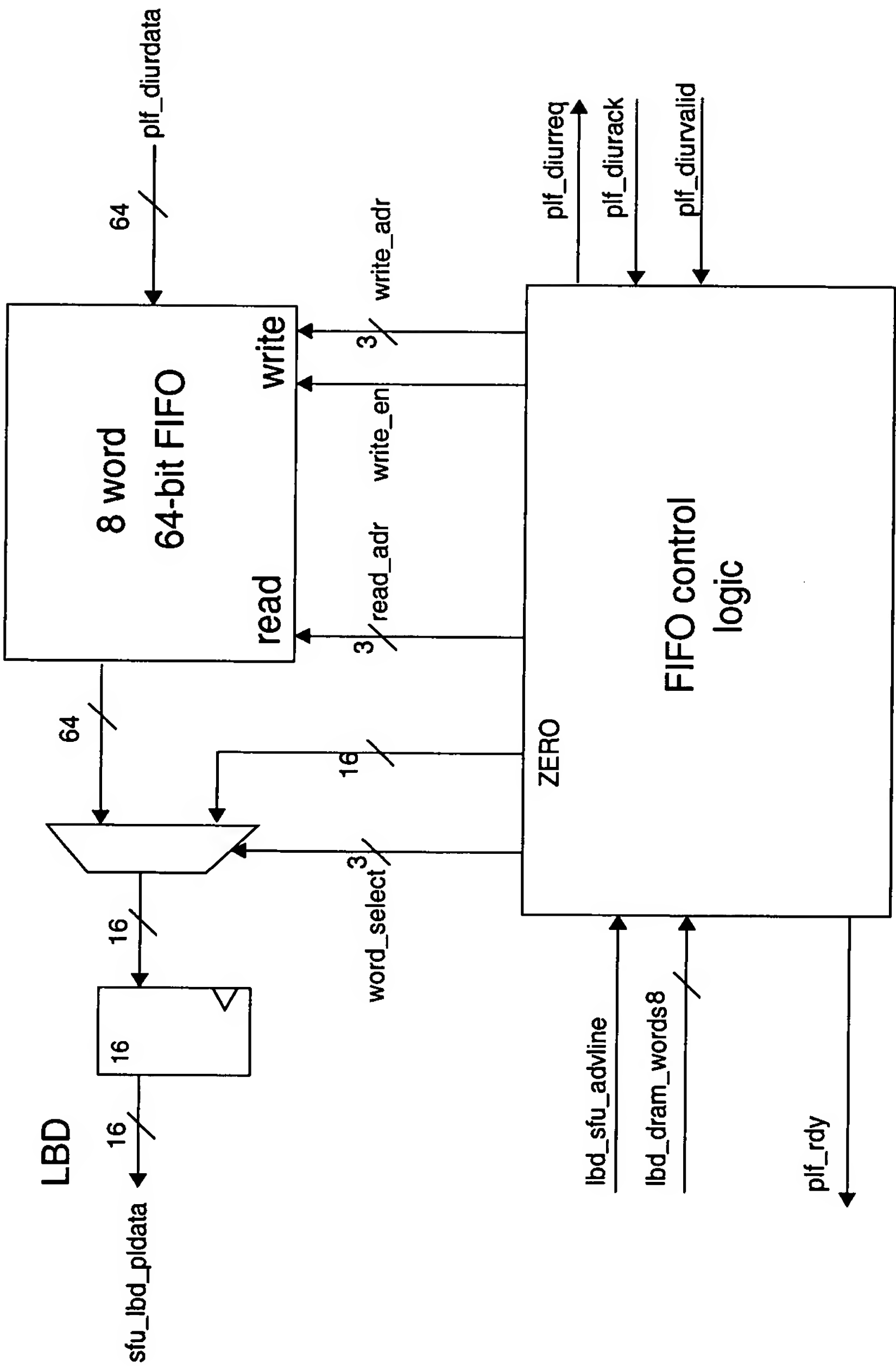


FIG. 161

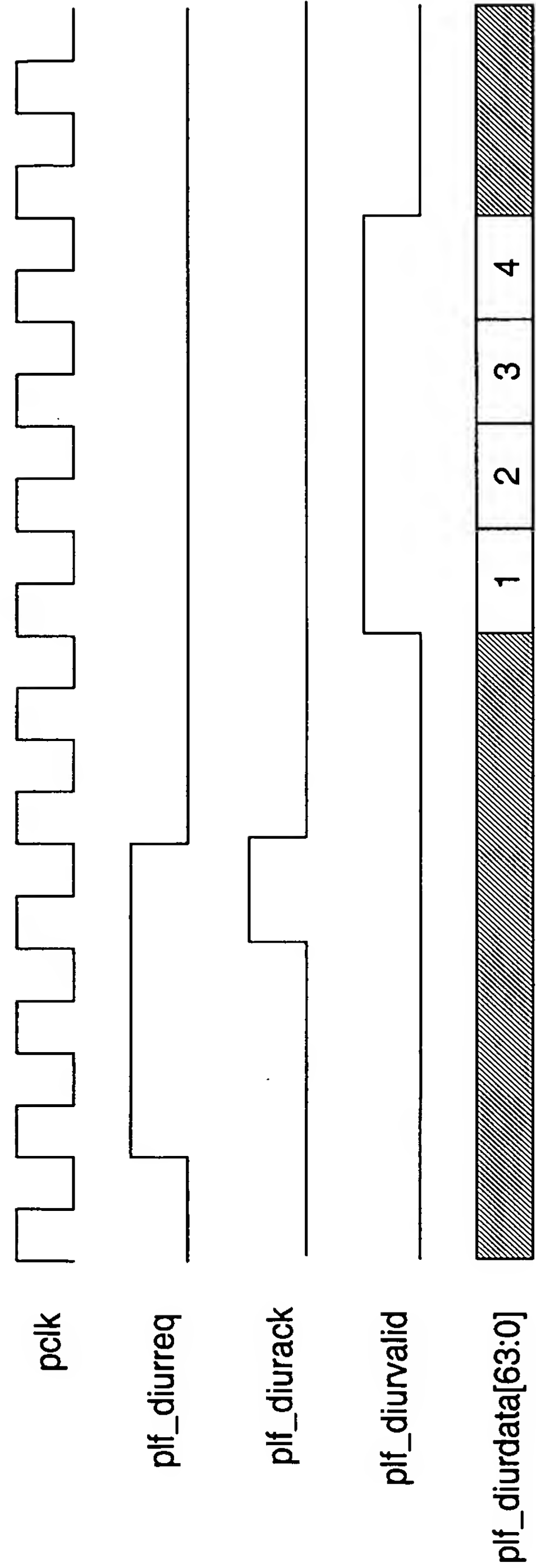


FIG. 162

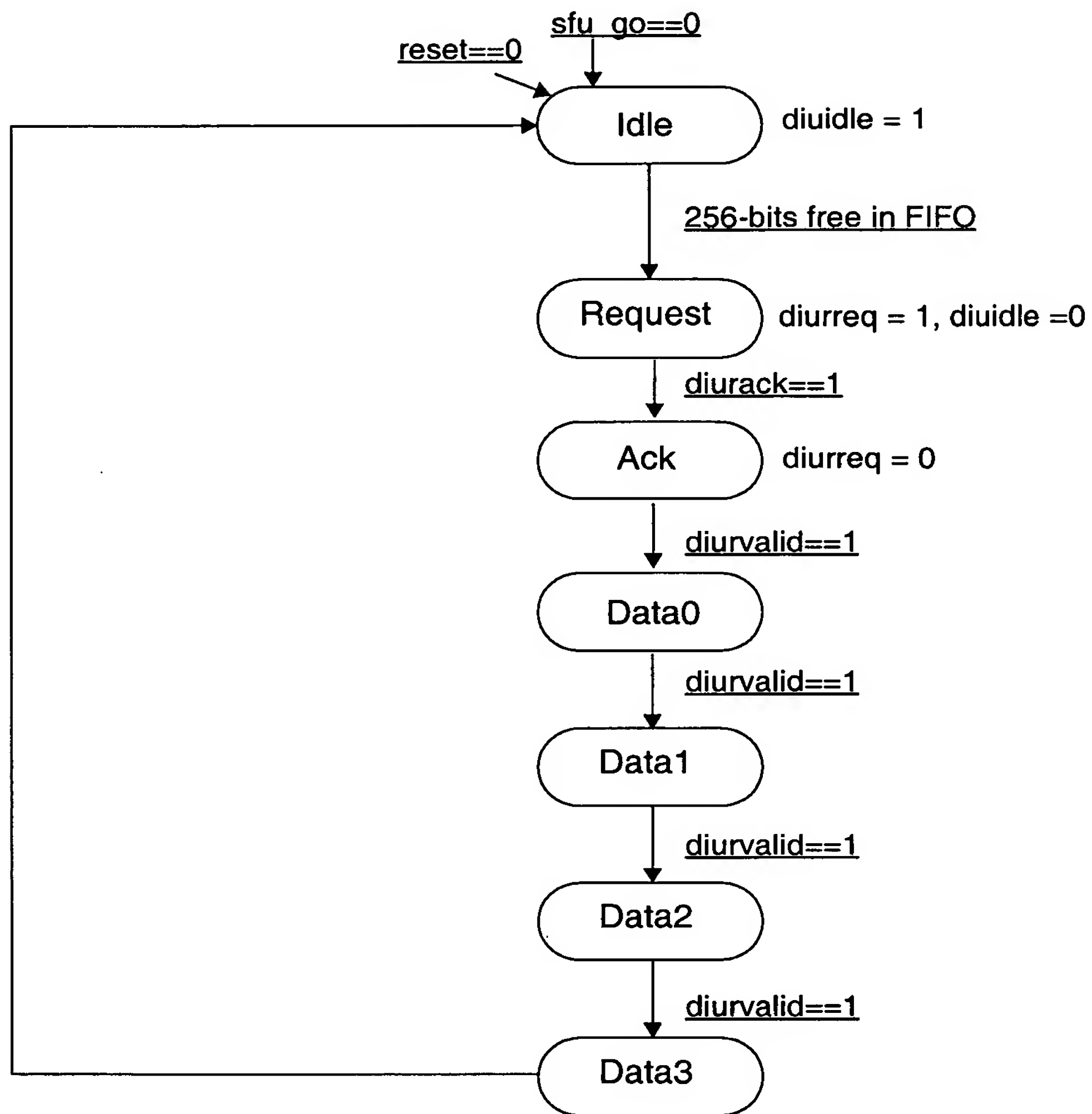


FIG. 163

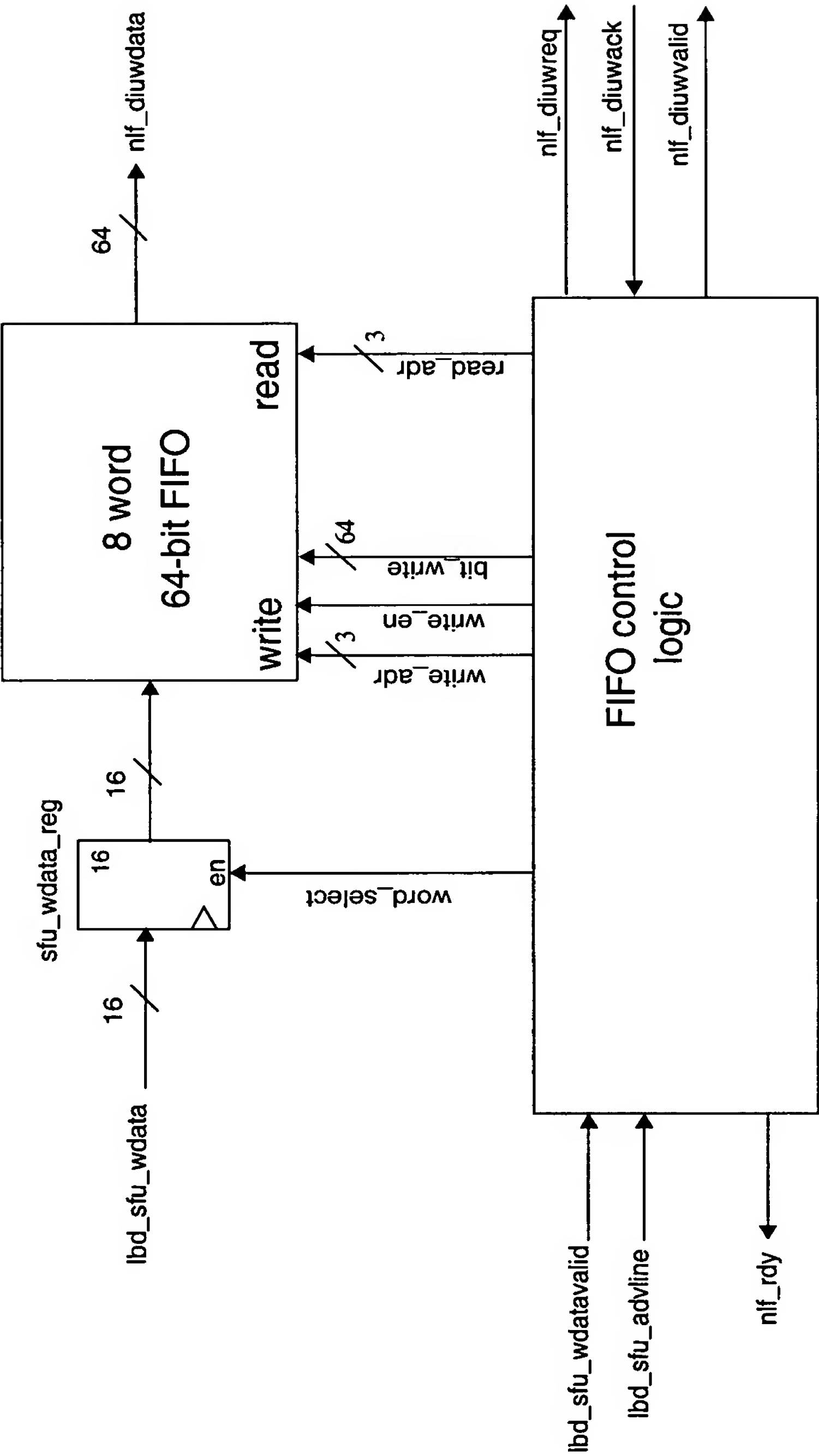


FIG. 164

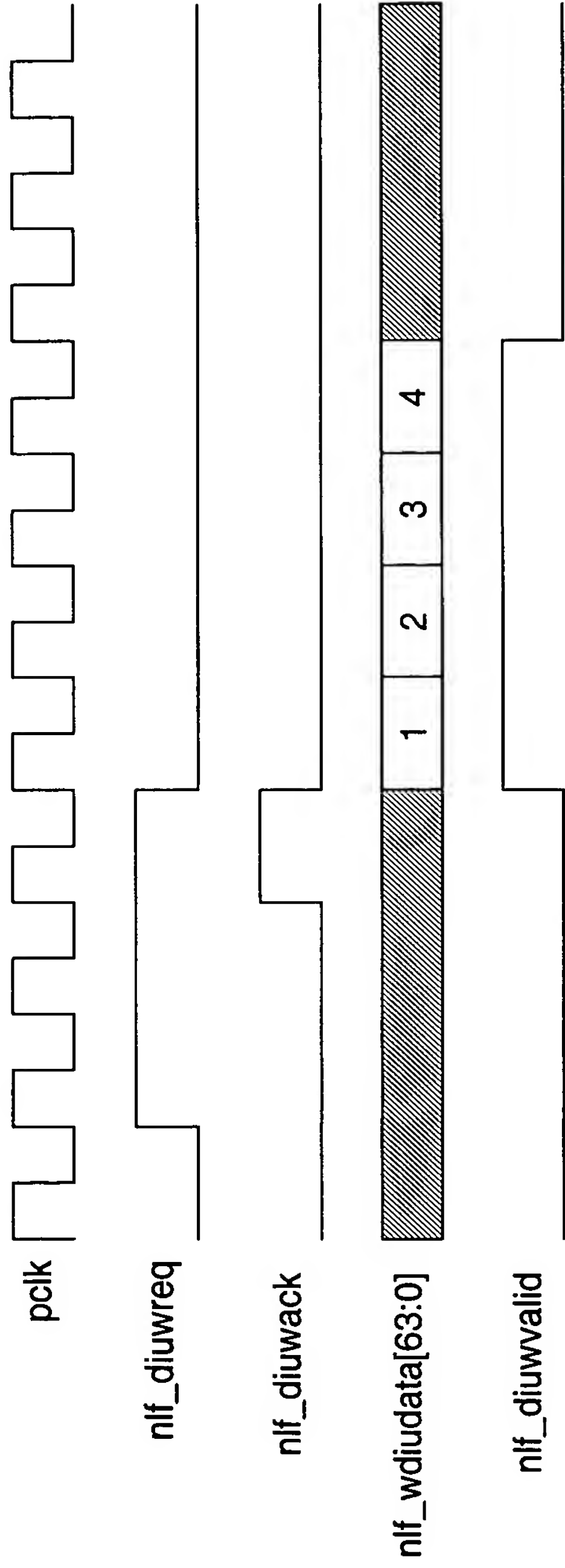


FIG. 165

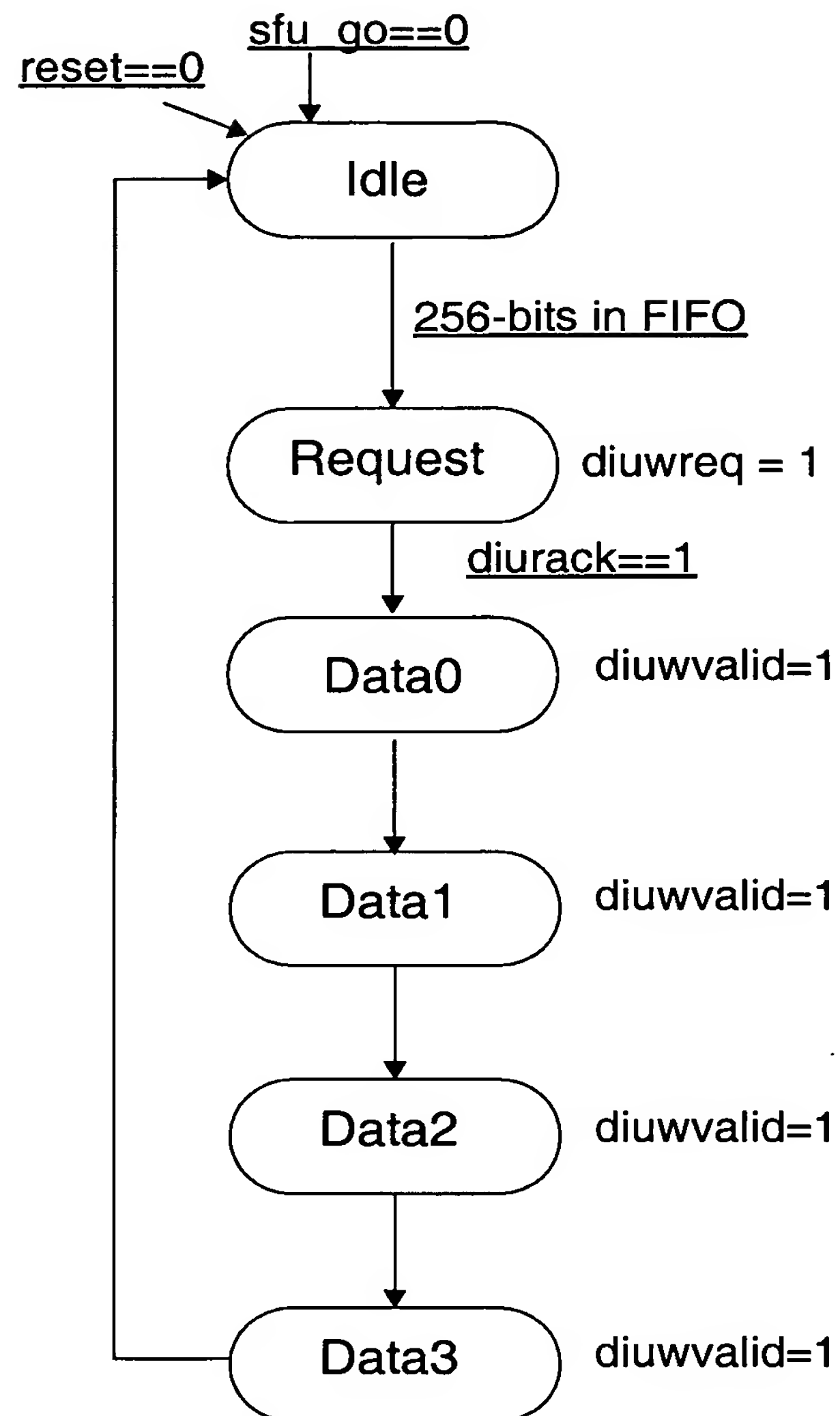


FIG. 166

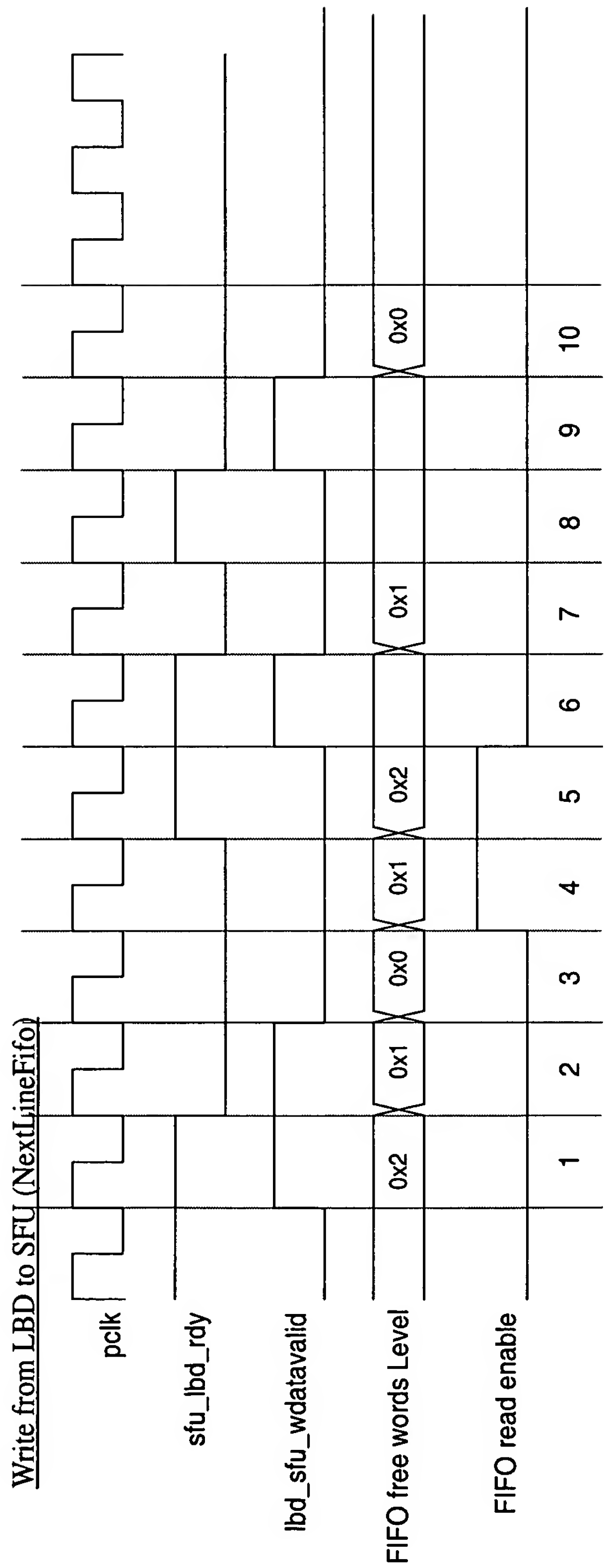


FIG. 167

Read from SFU to LBD (PreviousLineFifo)

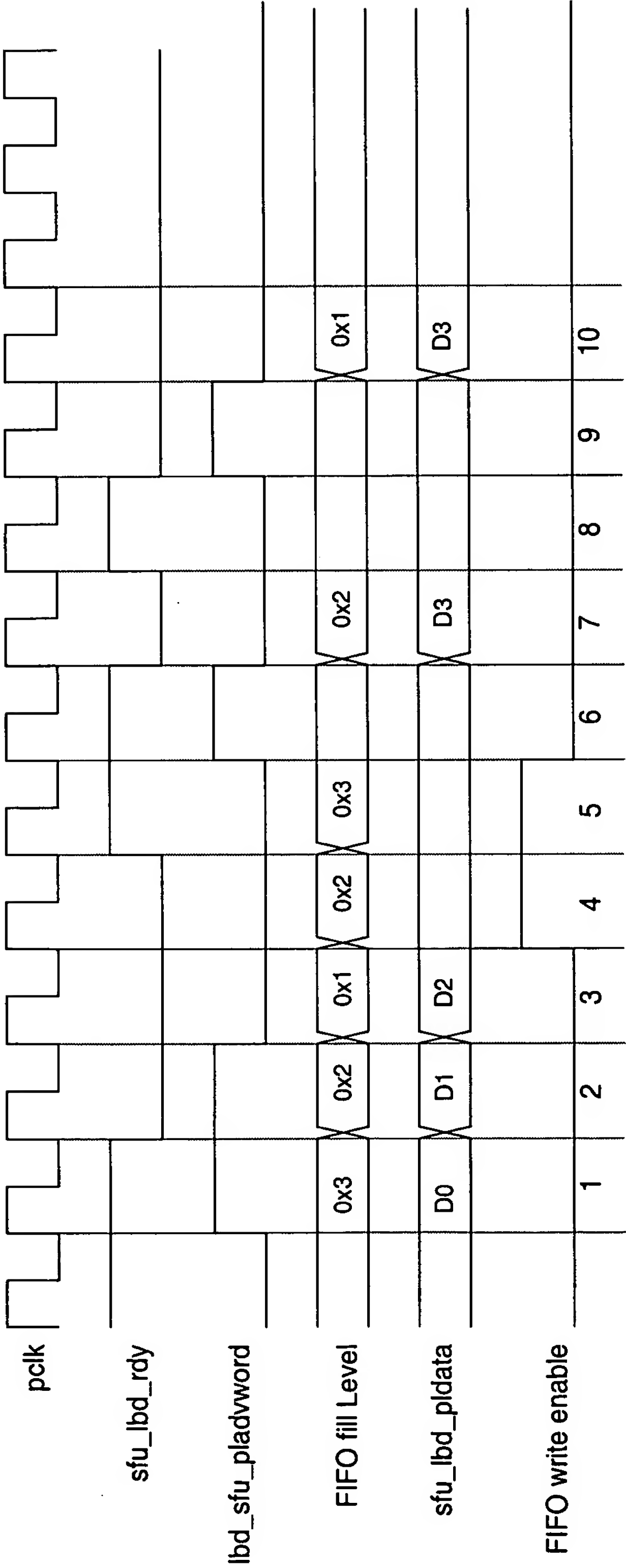


FIG. 168

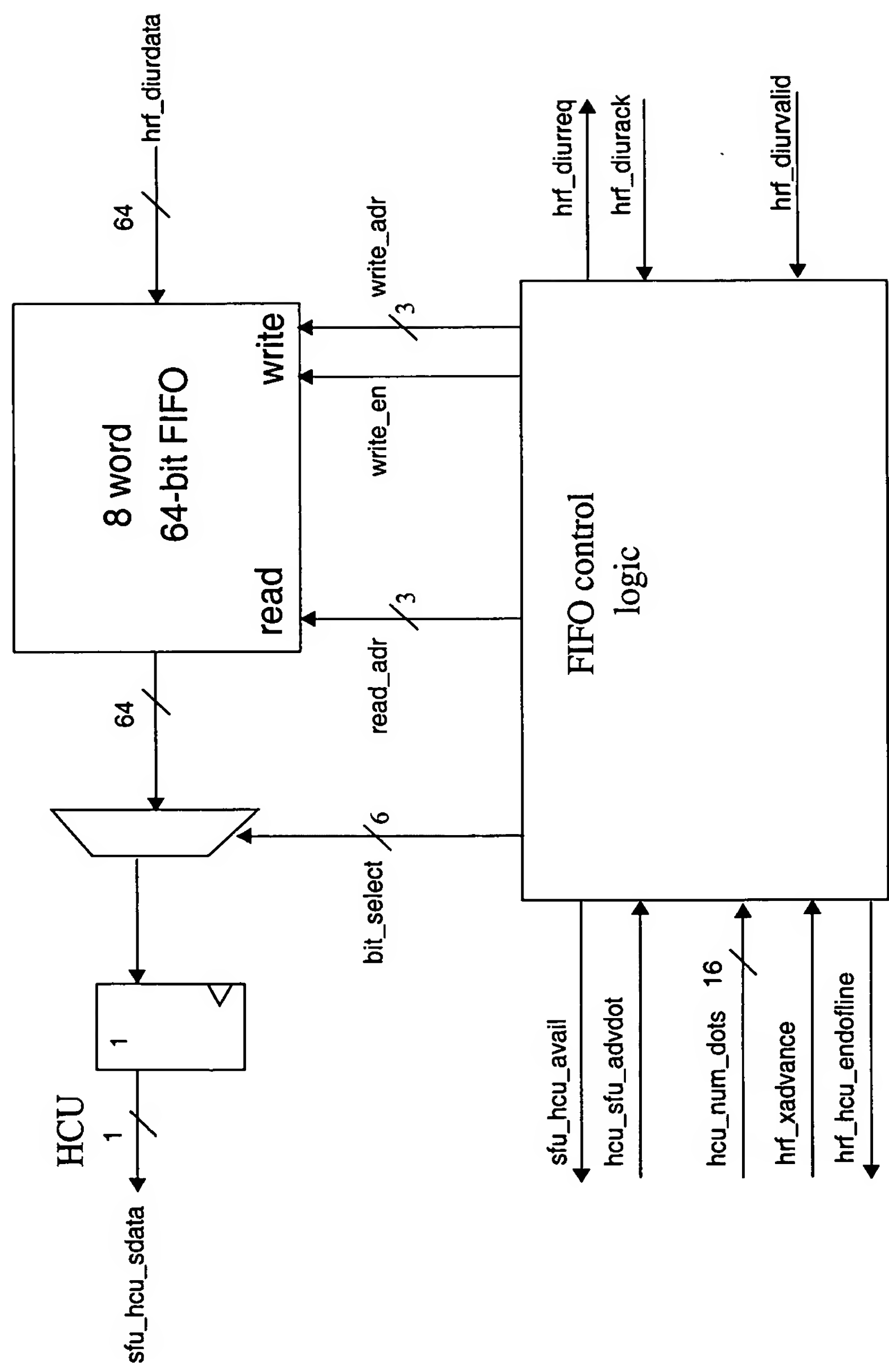


FIG. 169

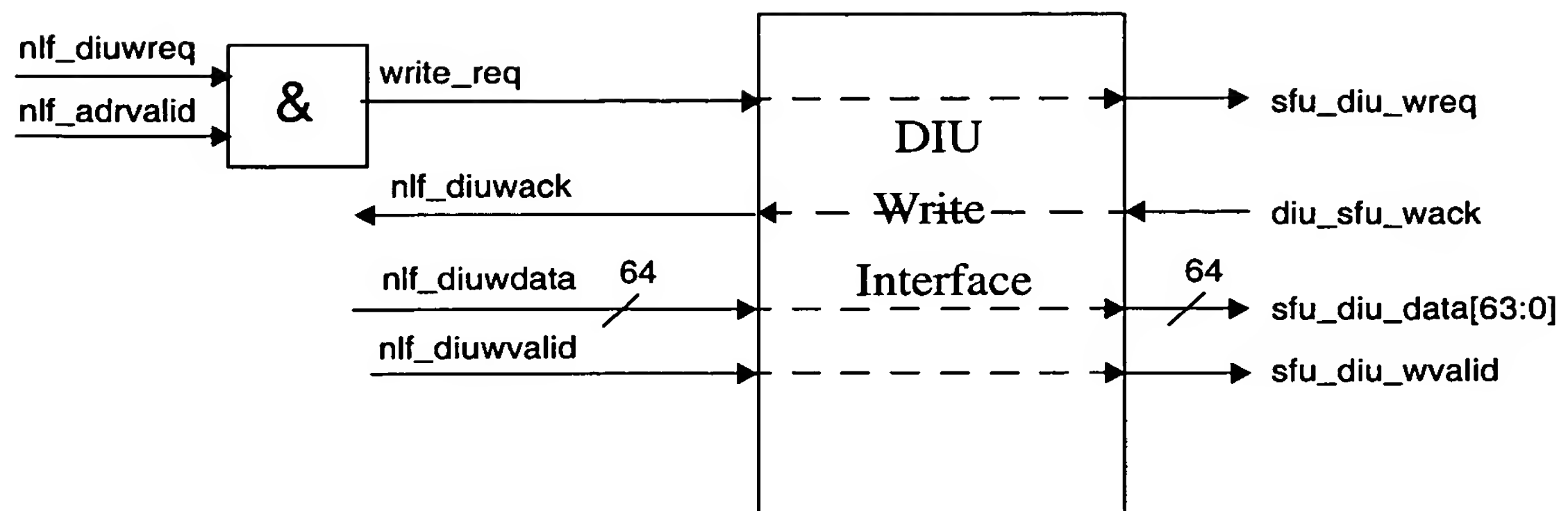


FIG. 170

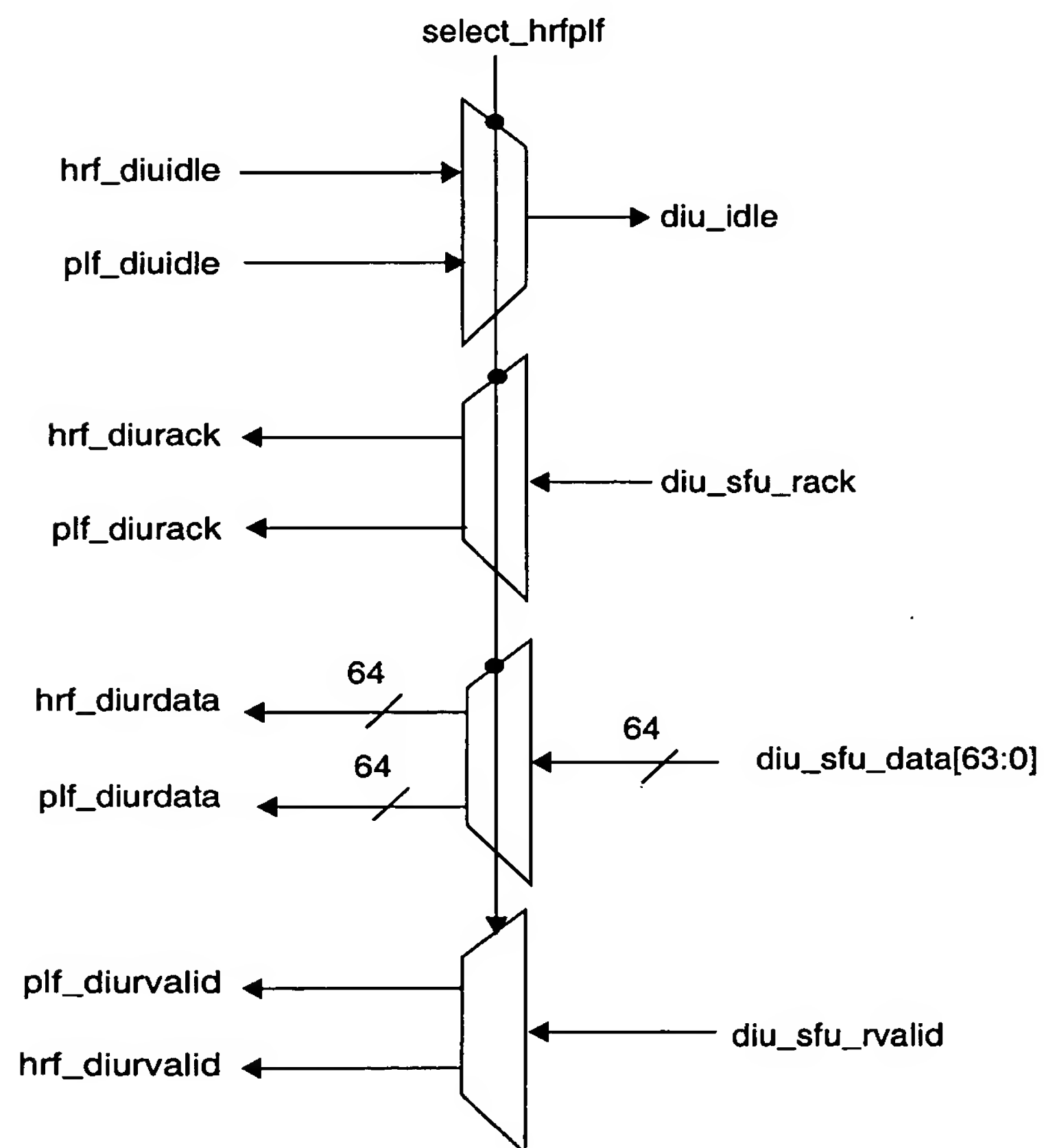


FIG. 171

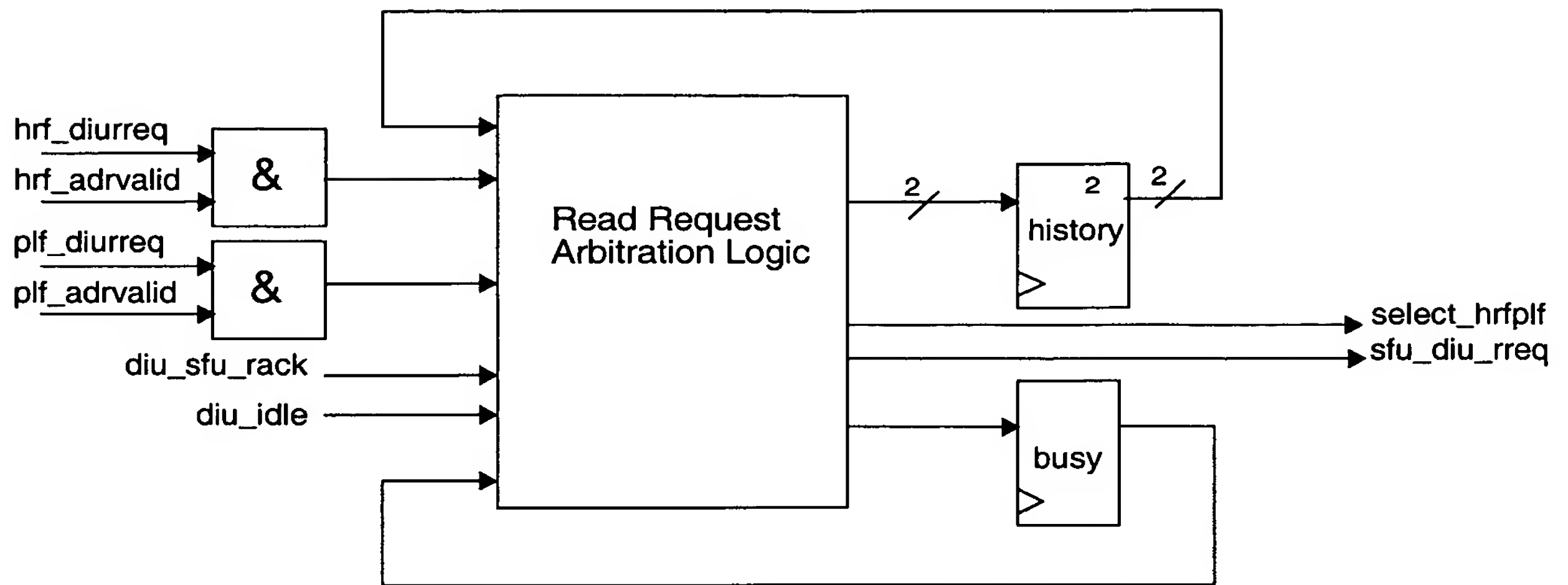


FIG. 172

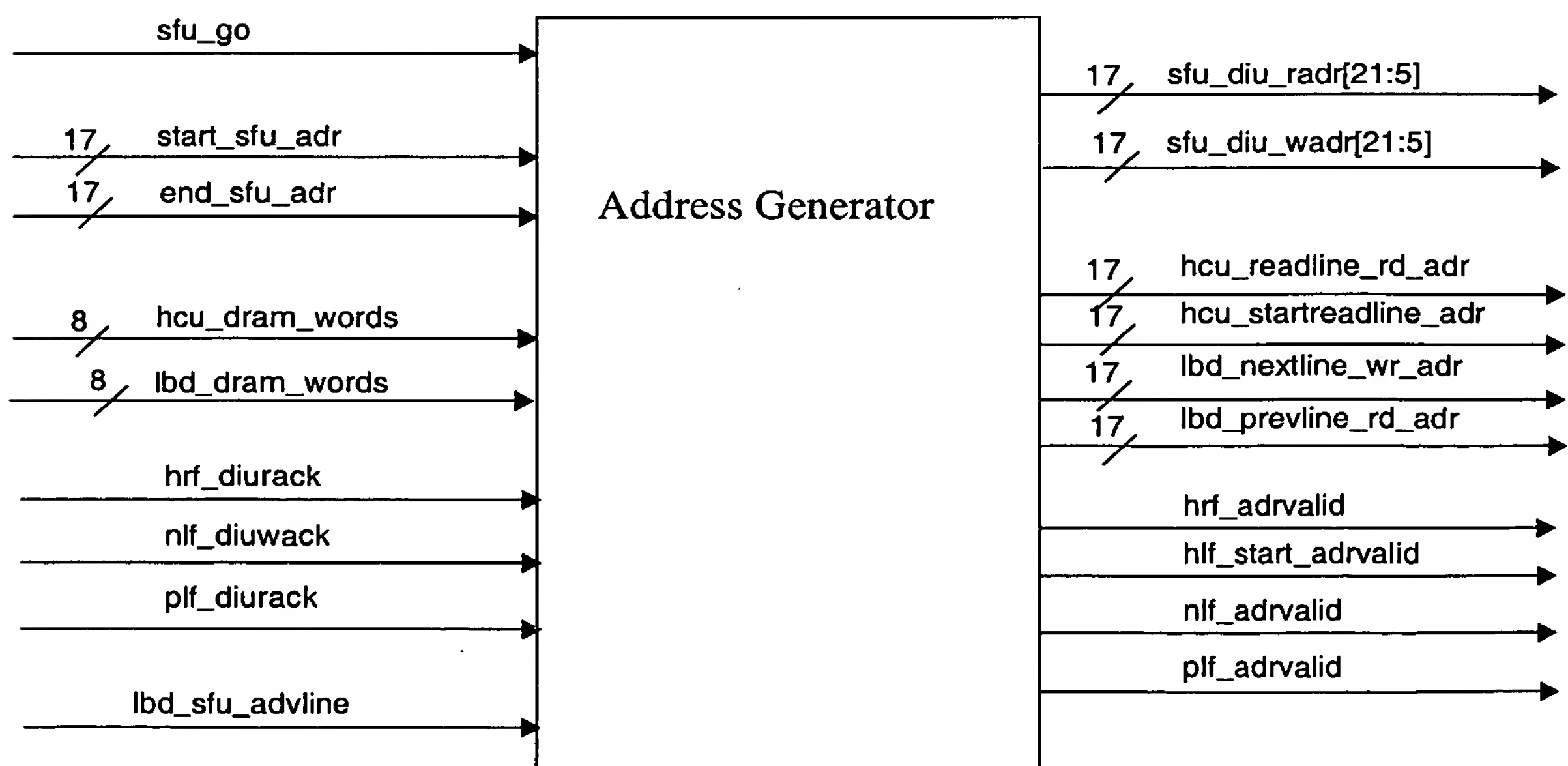
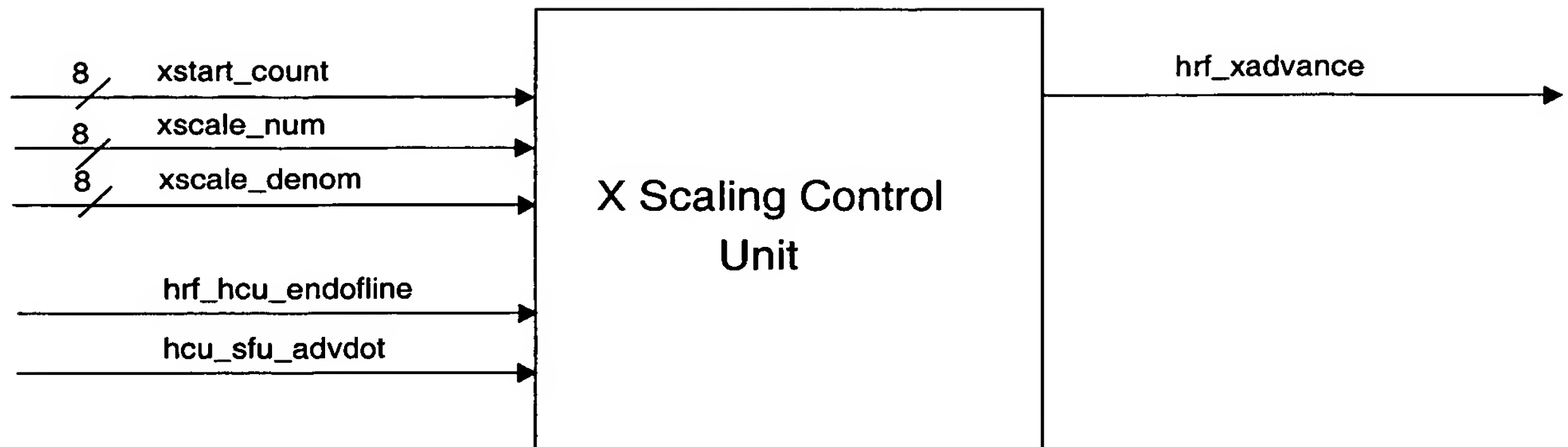
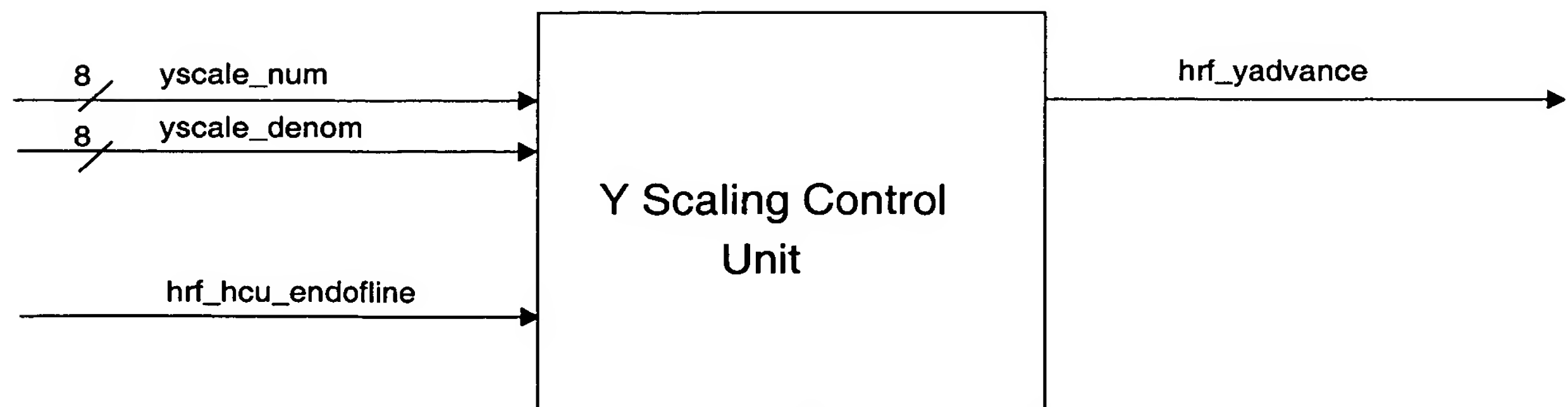


FIG. 173

*FIG. 174**FIG. 175*

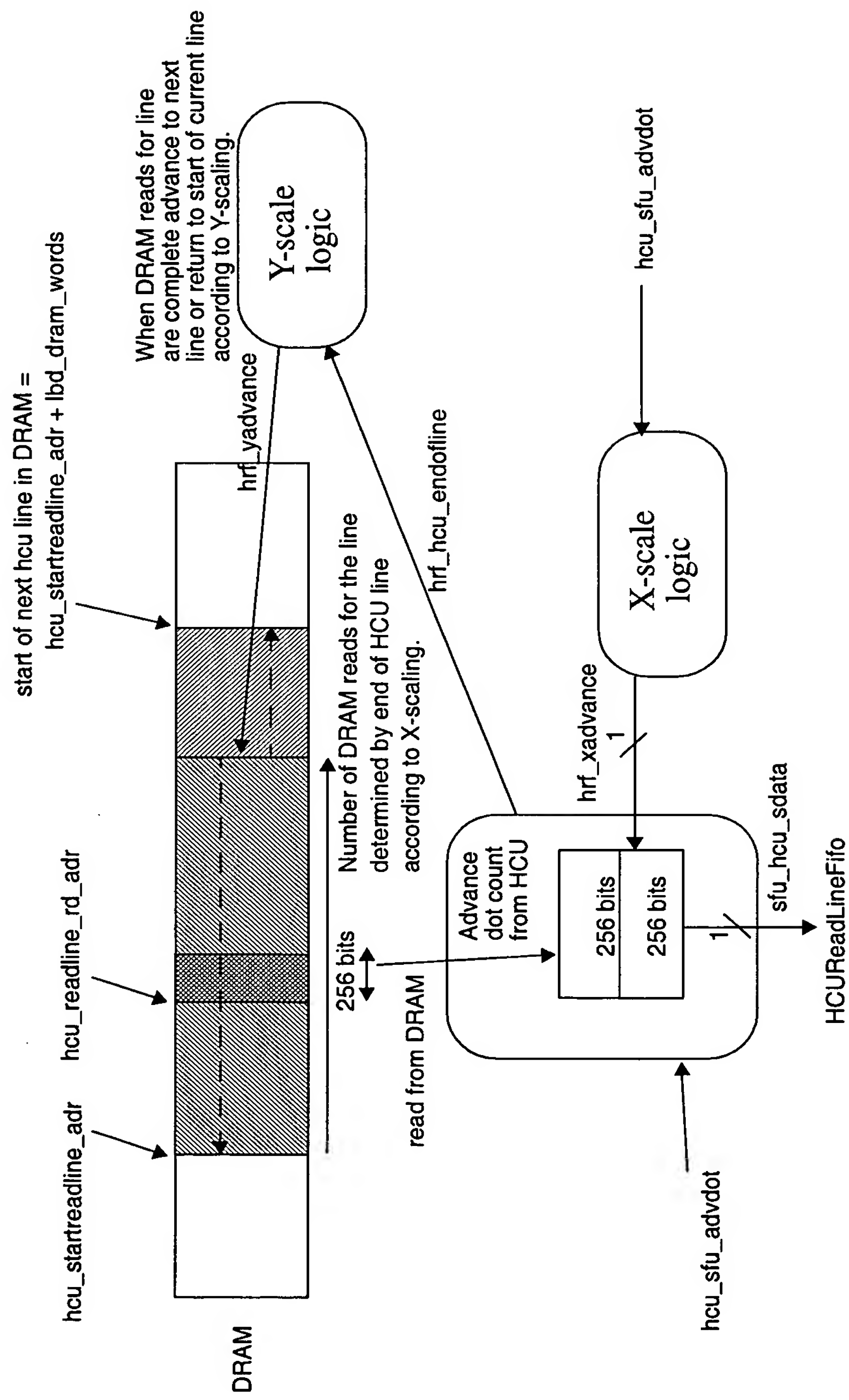


FIG. 176

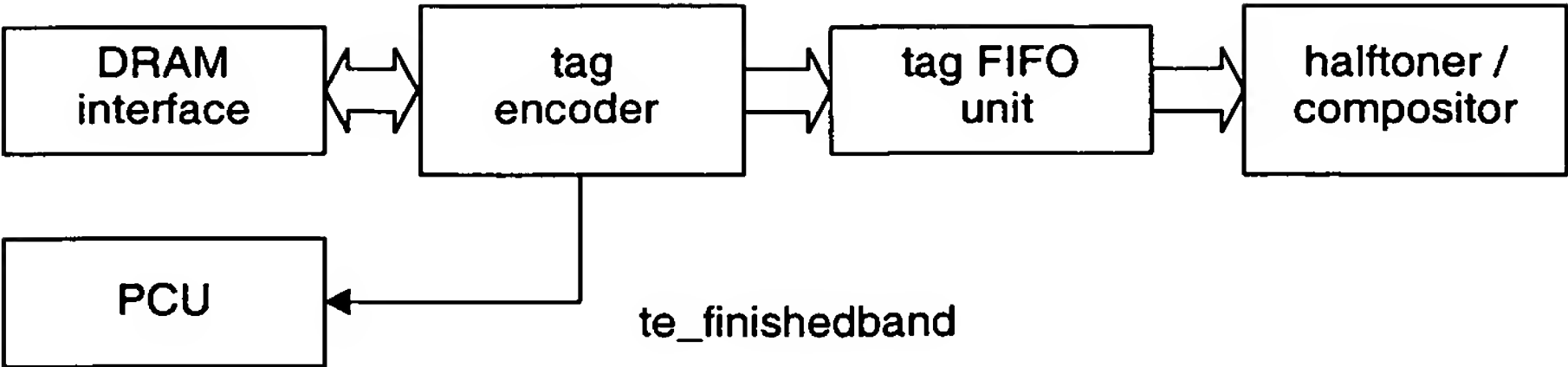


FIG. 177

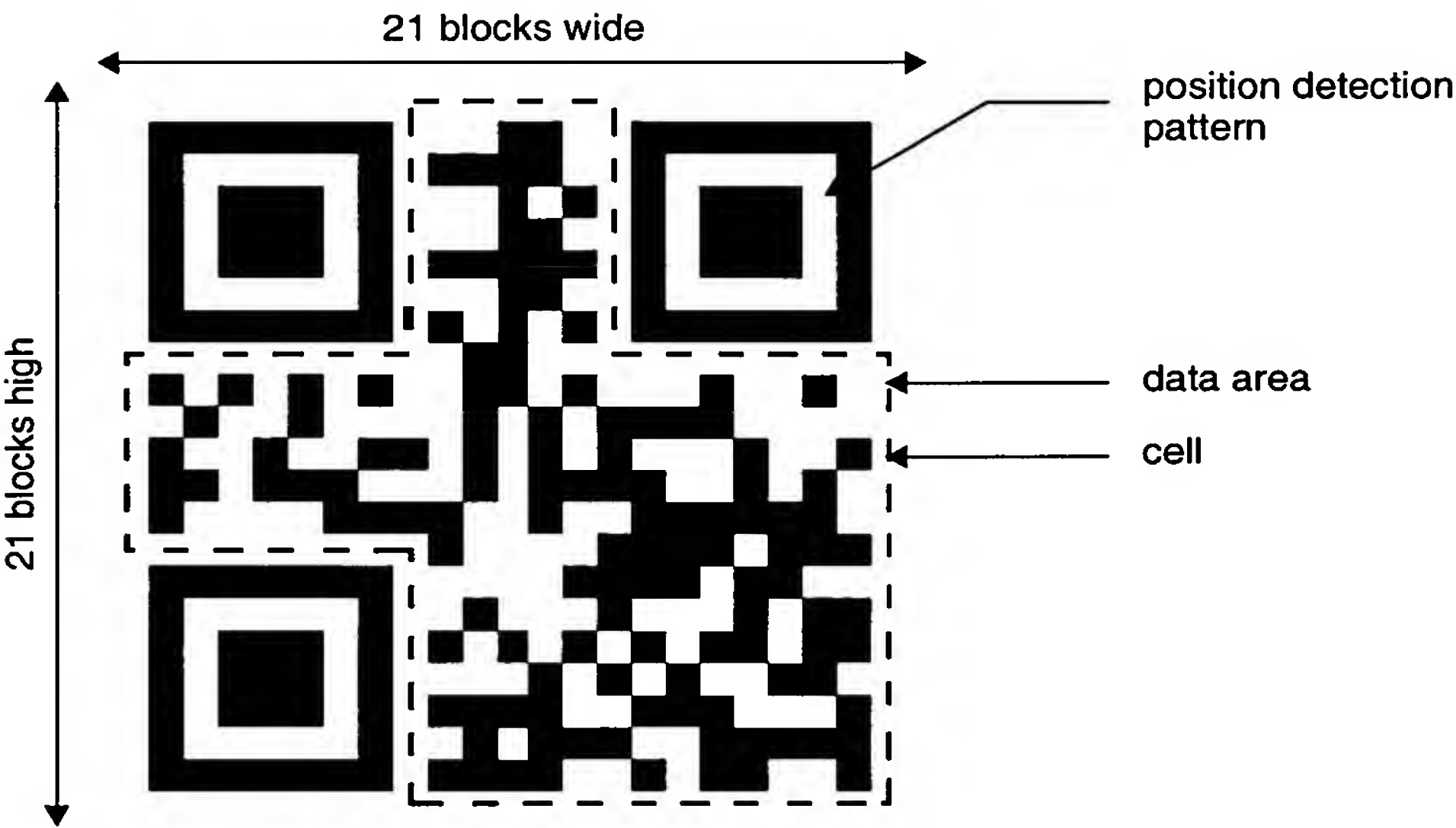
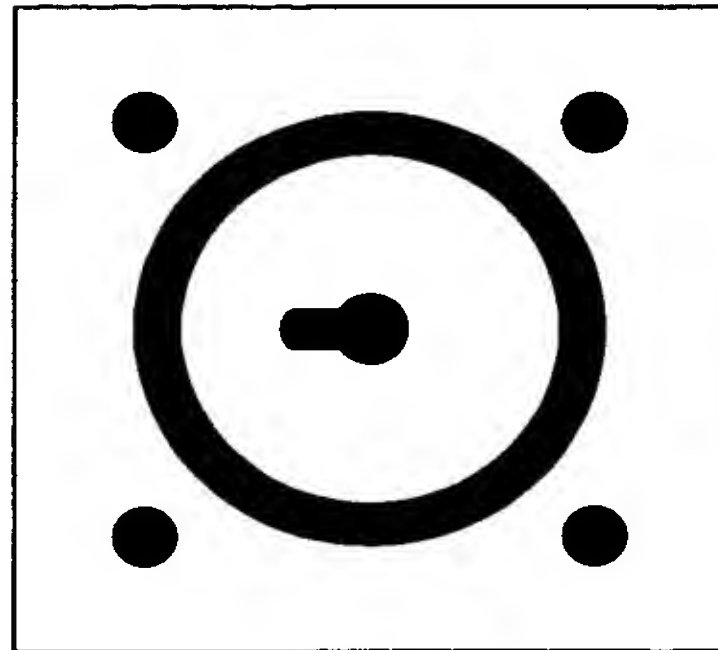
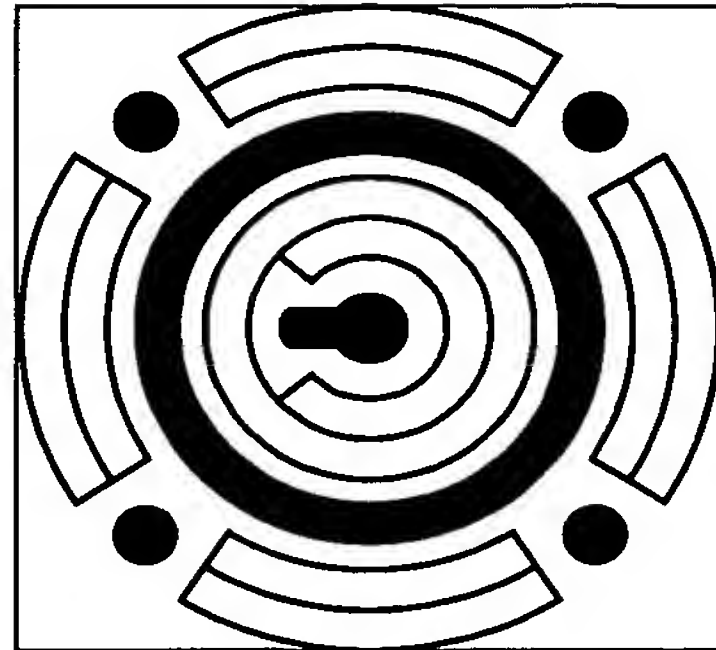


FIG. 178



(a) Netpage tag background pattern



(b) Netpage tag showing data area

FIG. 179

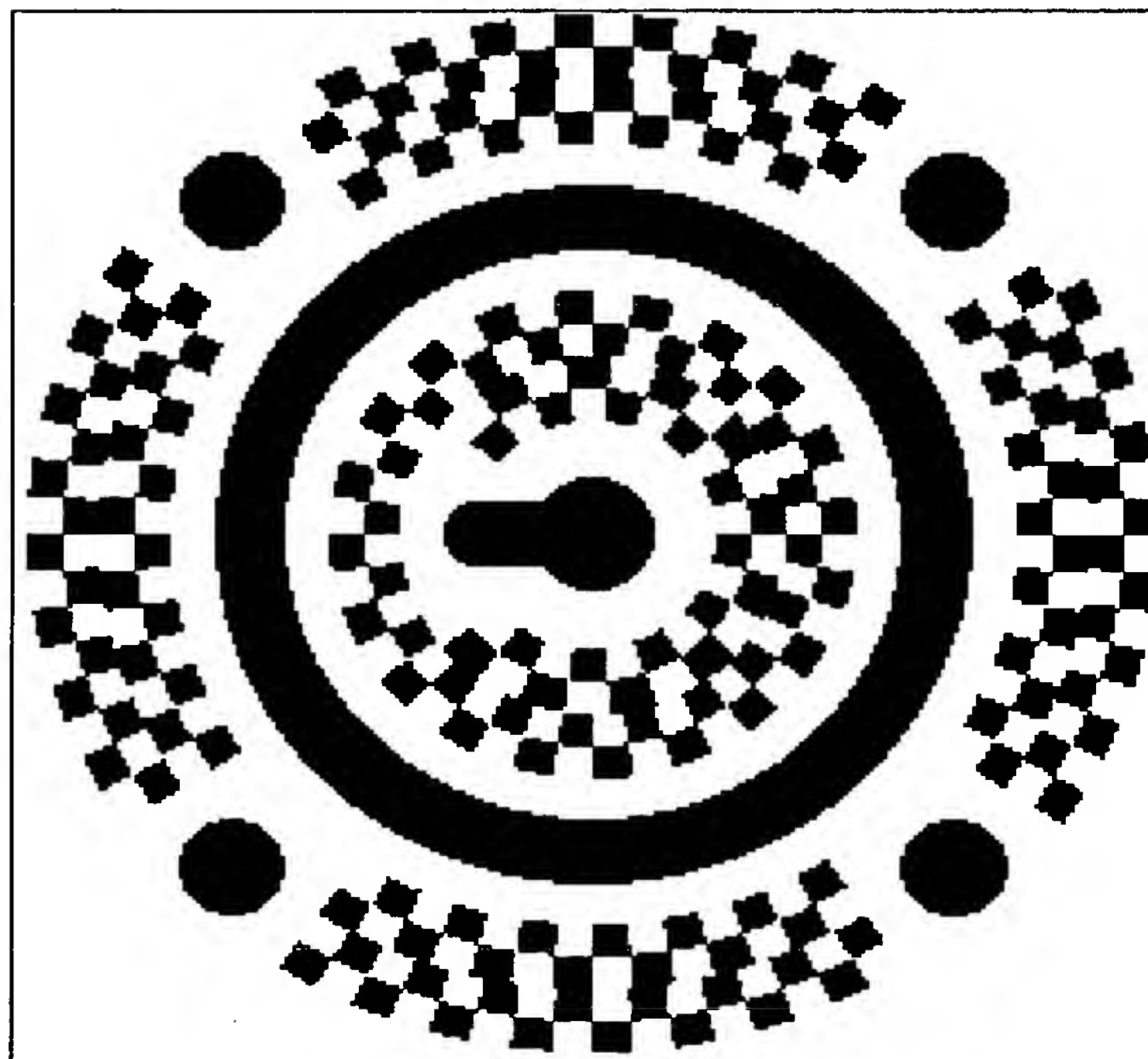


FIG. 180

156/331

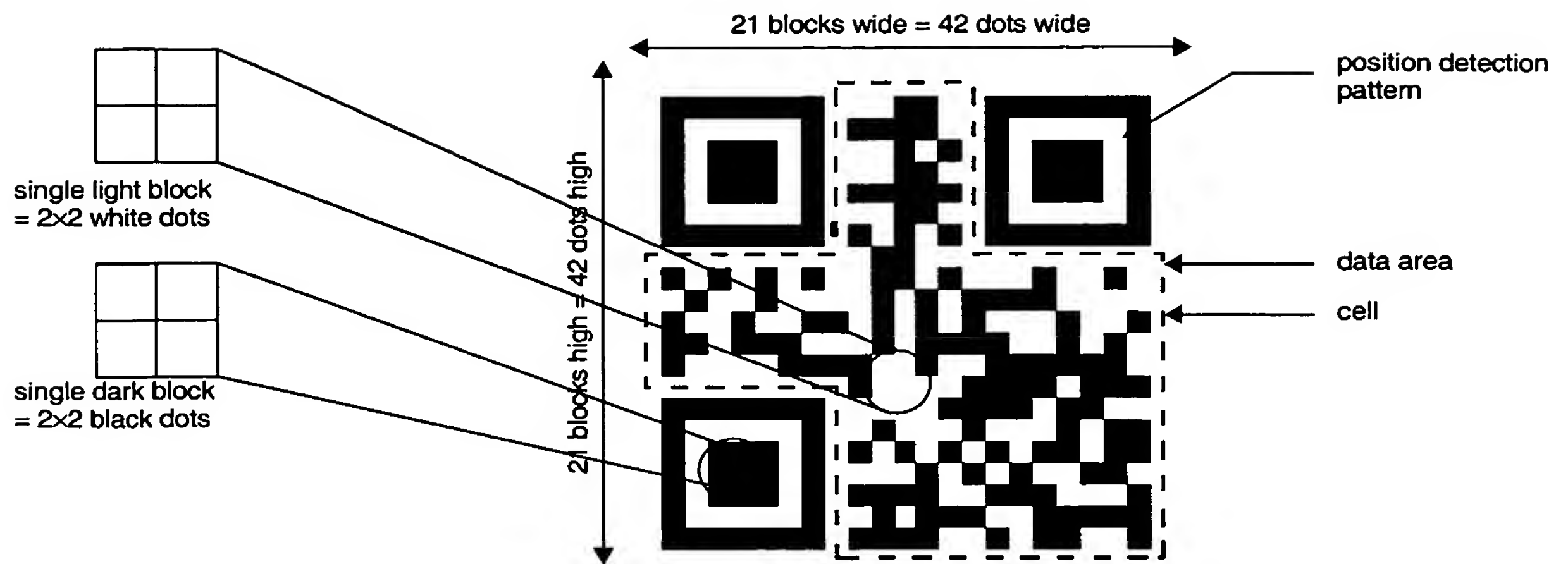


FIG. 181

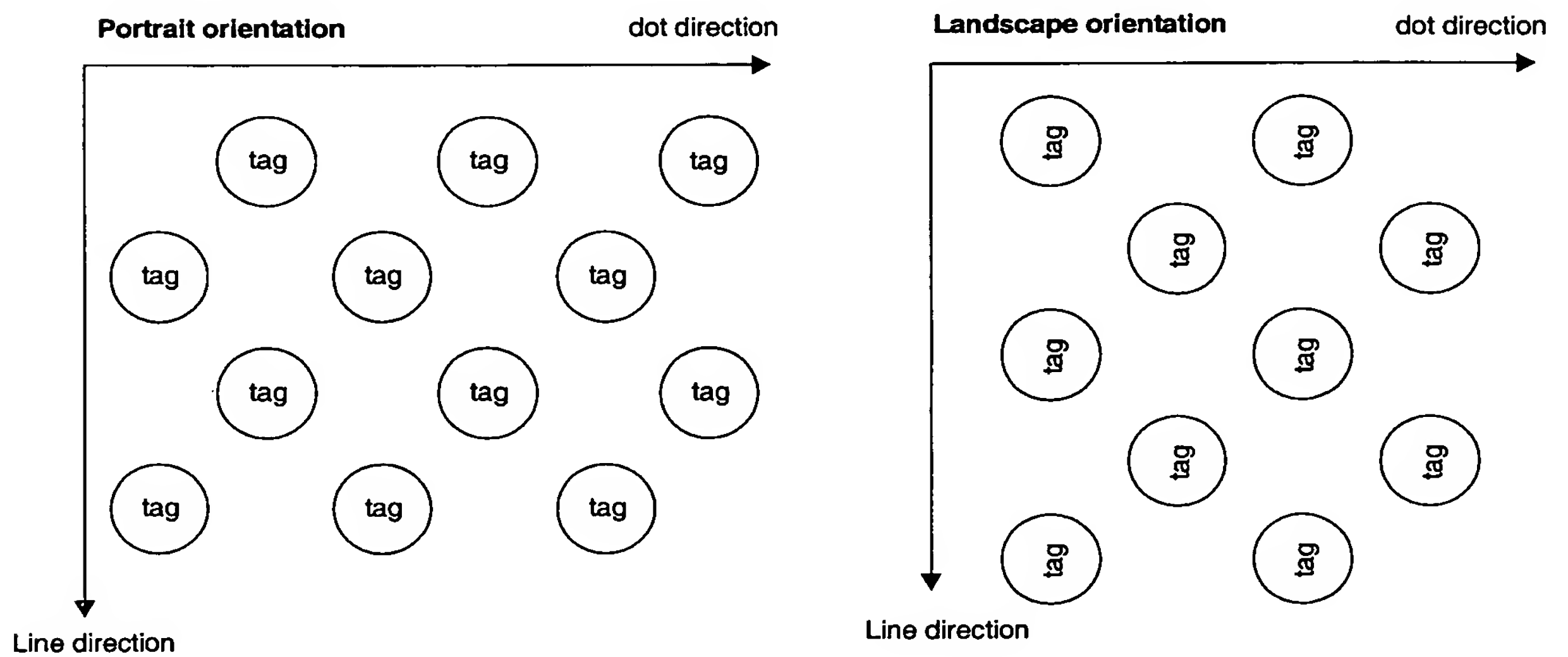


FIG. 182

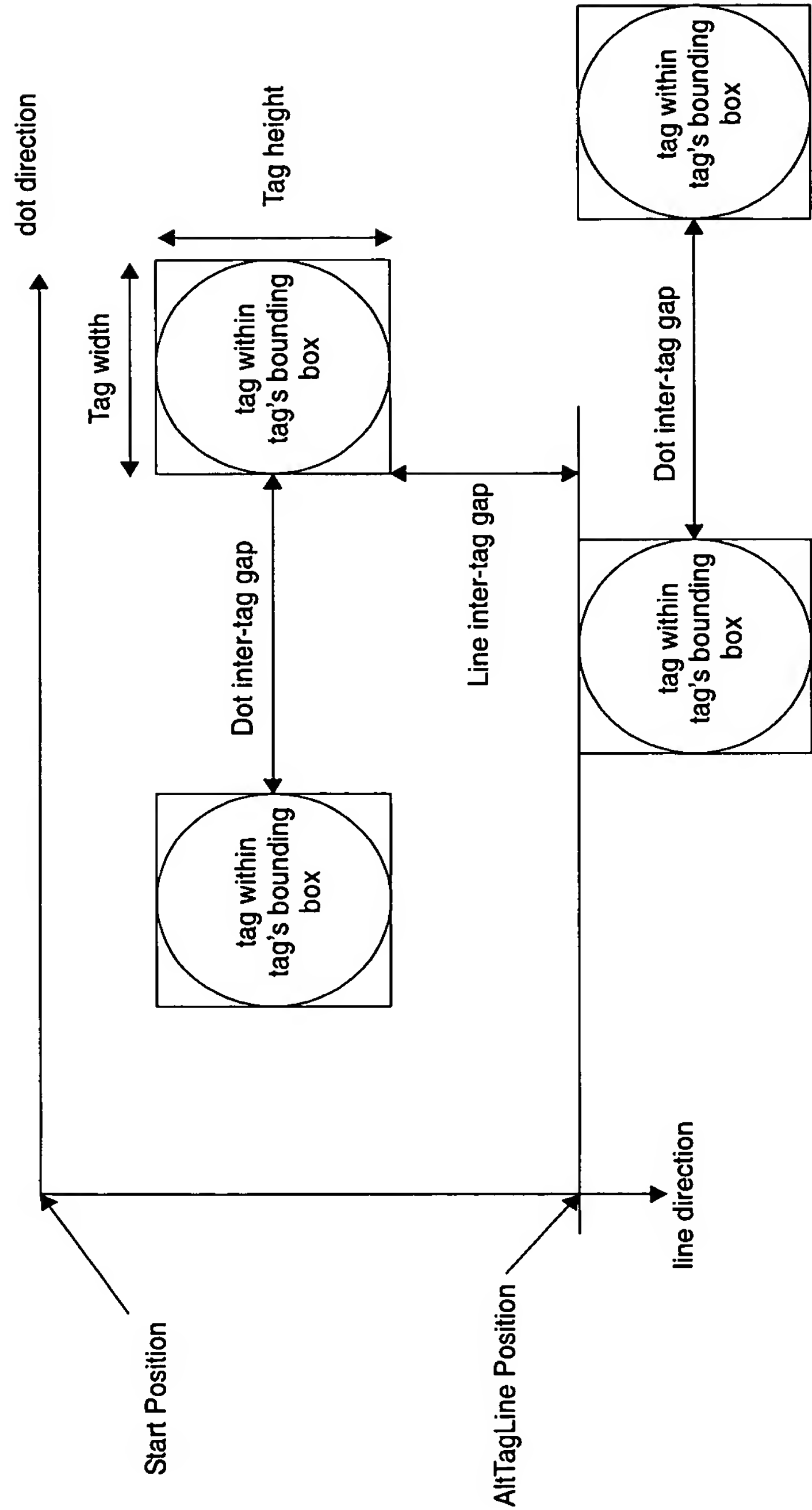


FIG. 183

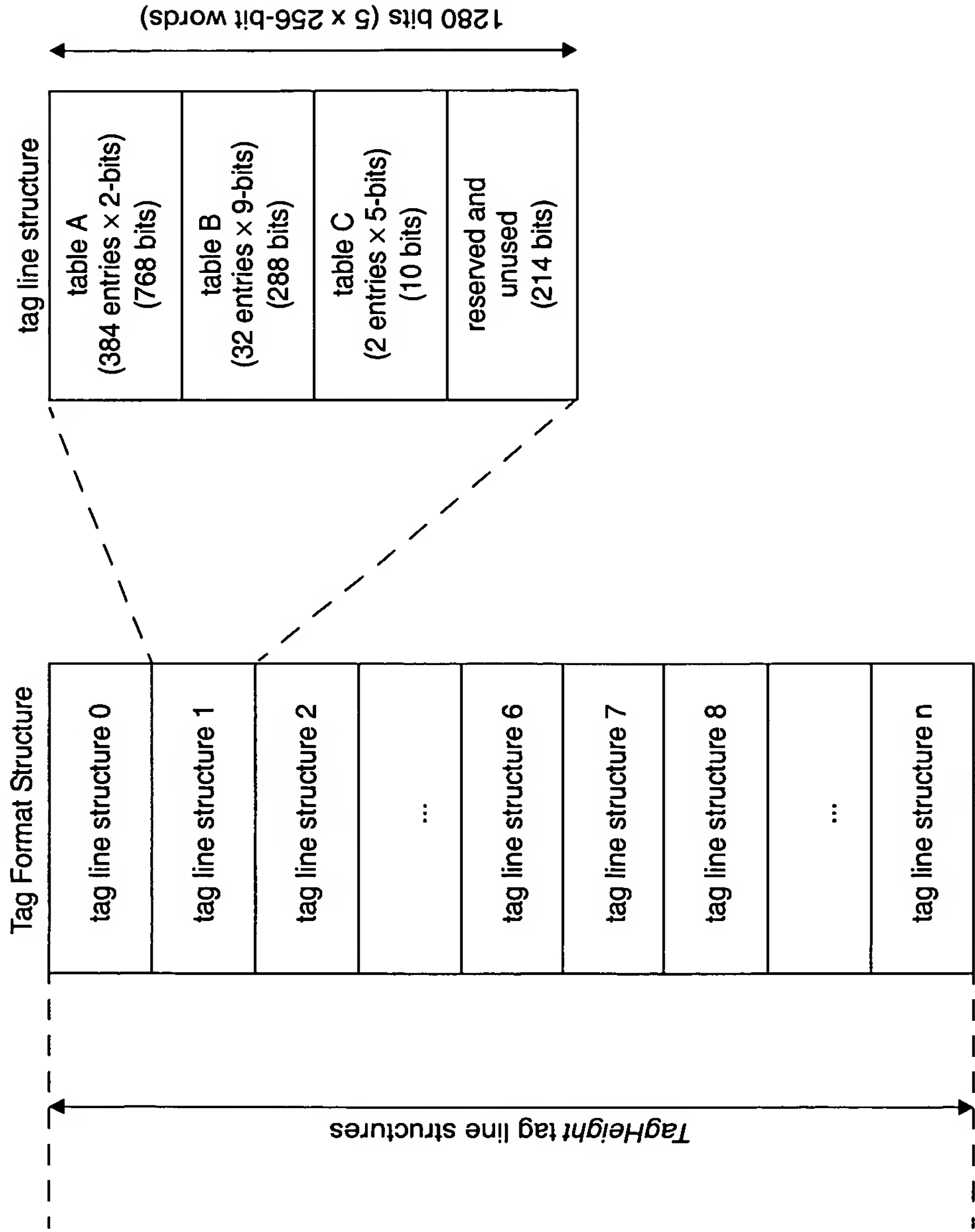


FIG. 184

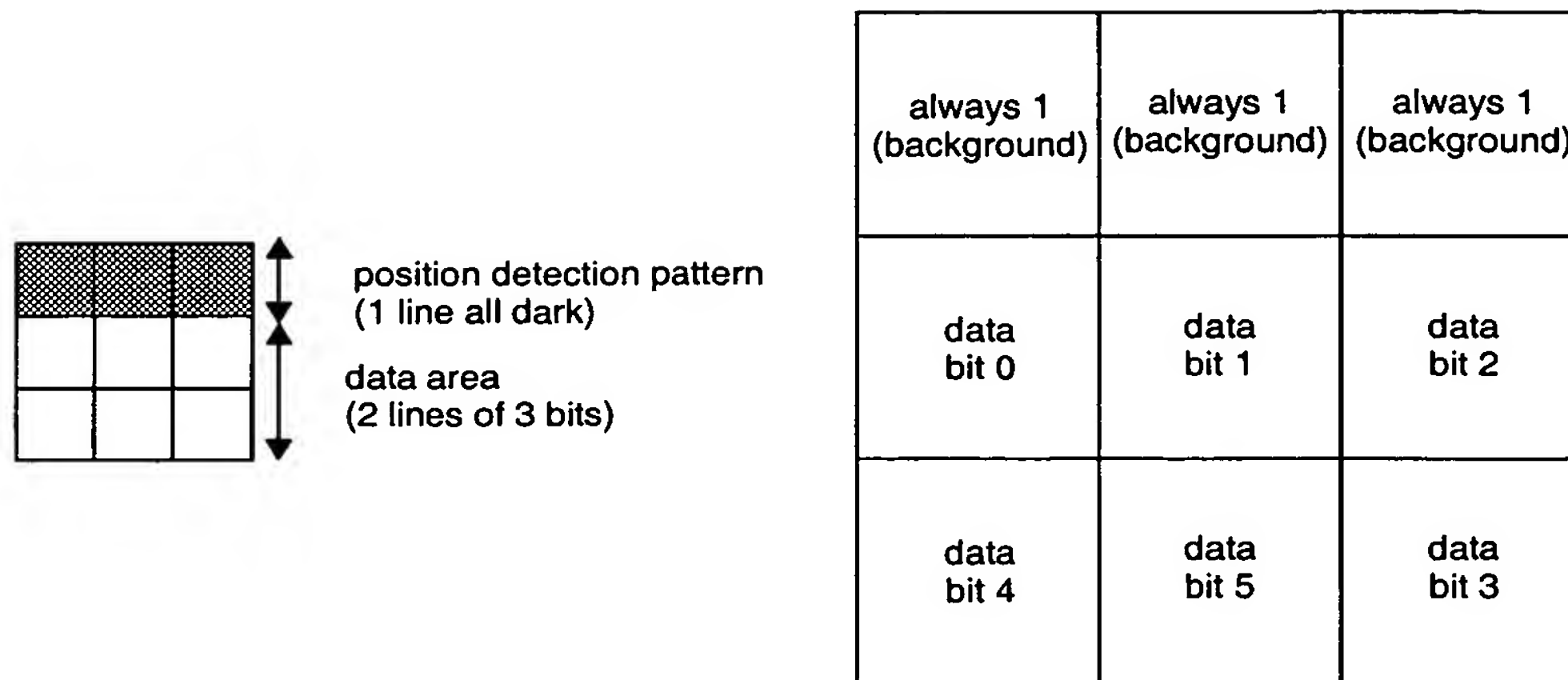


FIG. 185

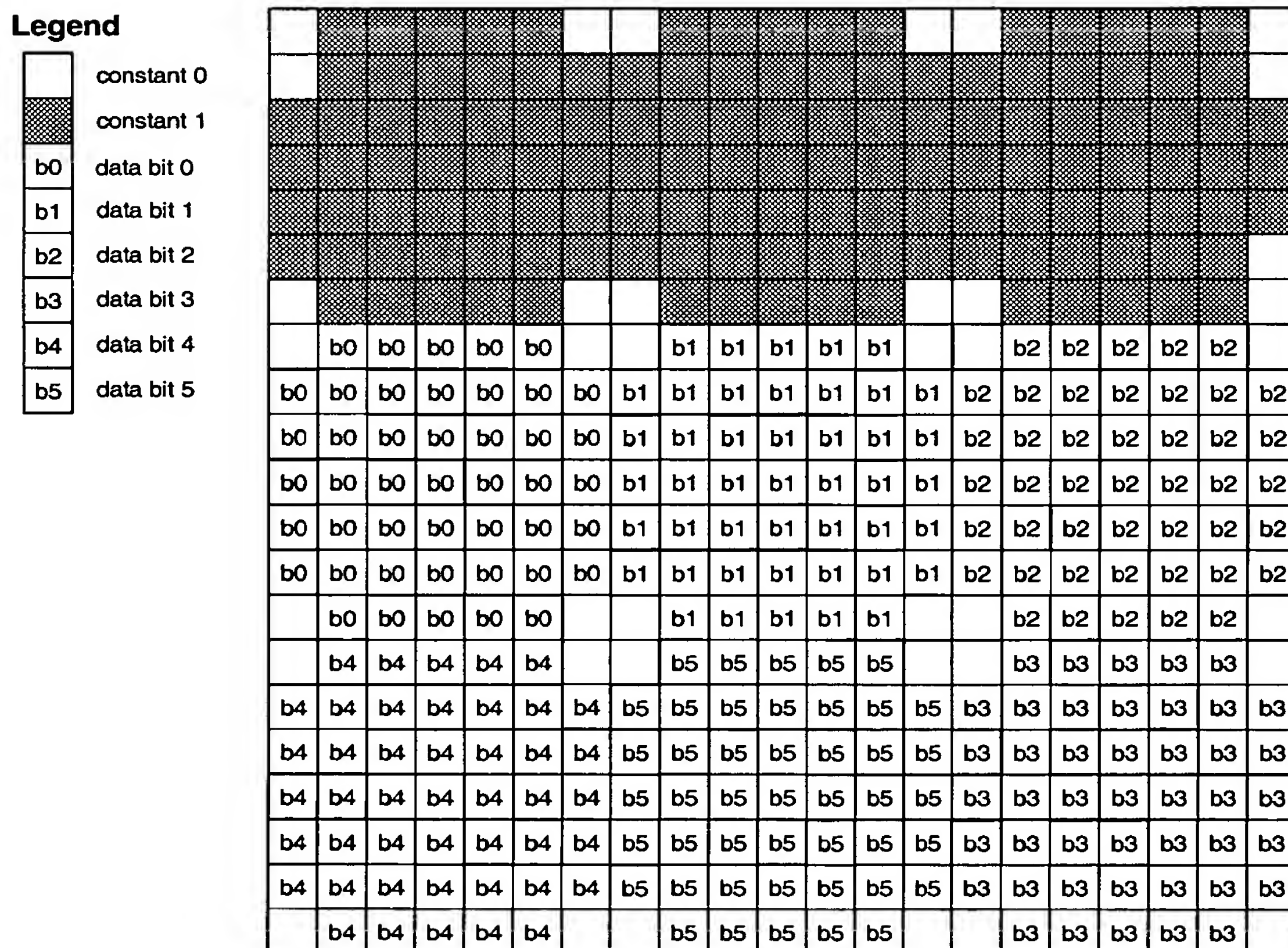


FIG. 186

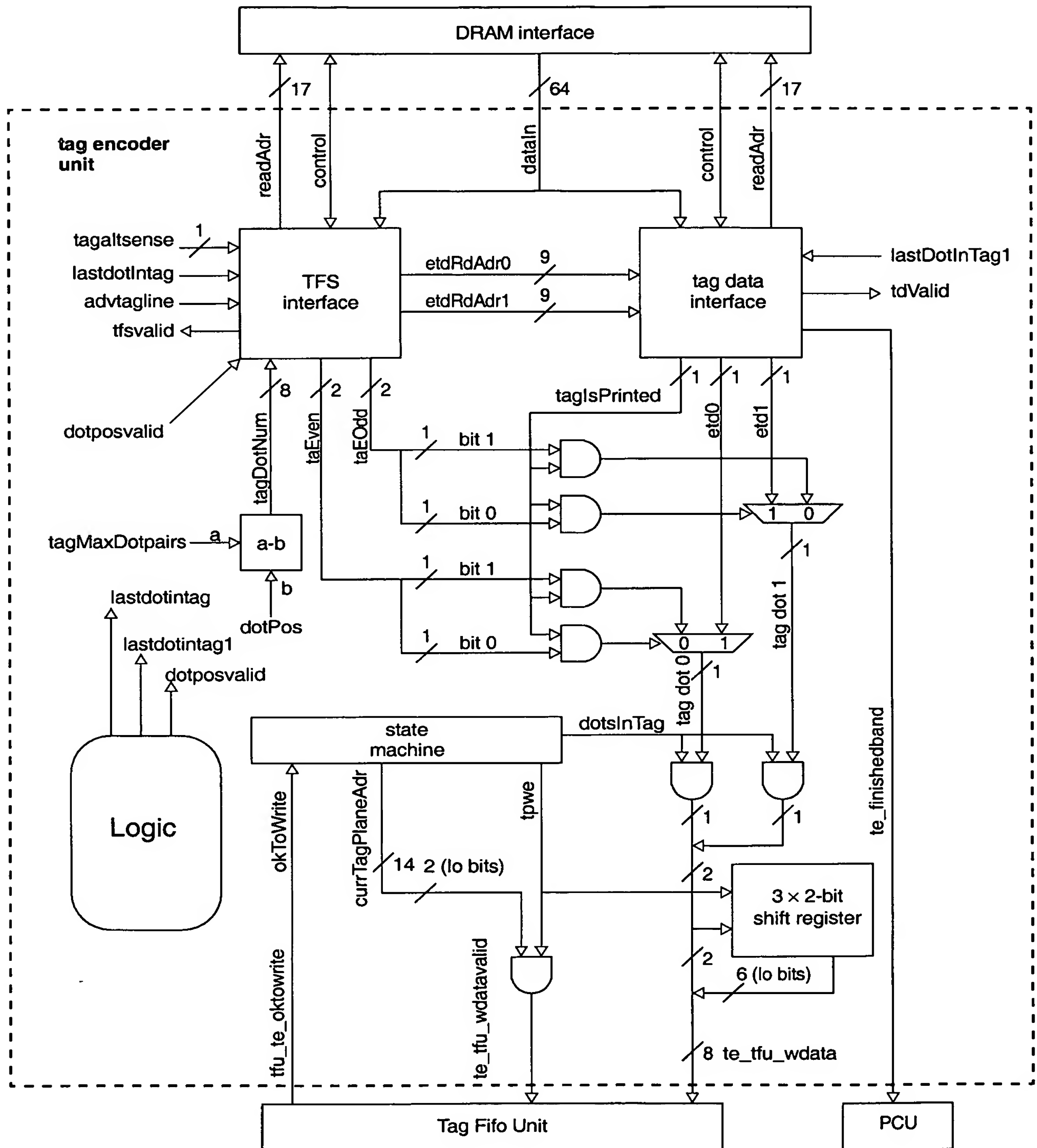


FIG. 187

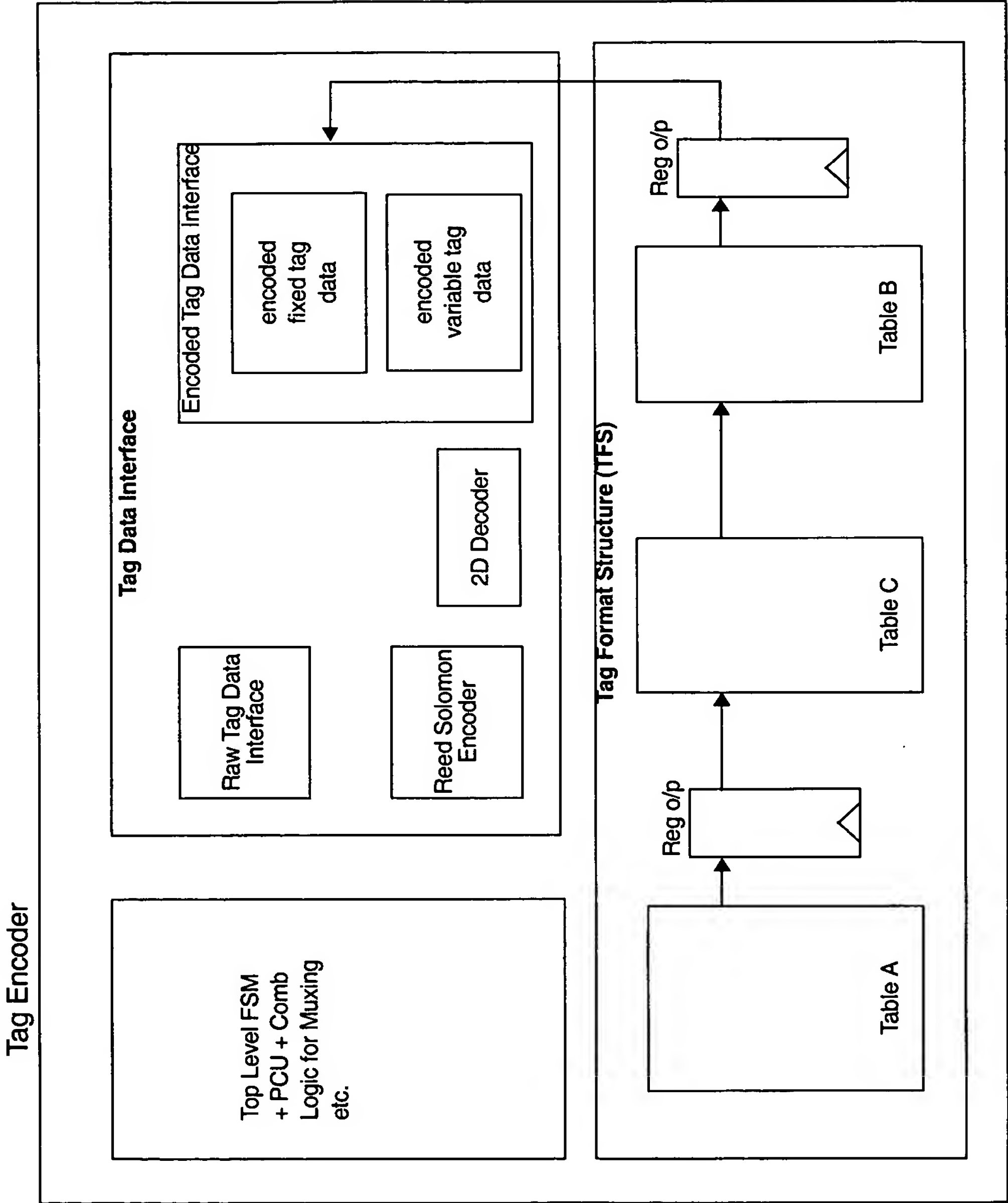


FIG. 188

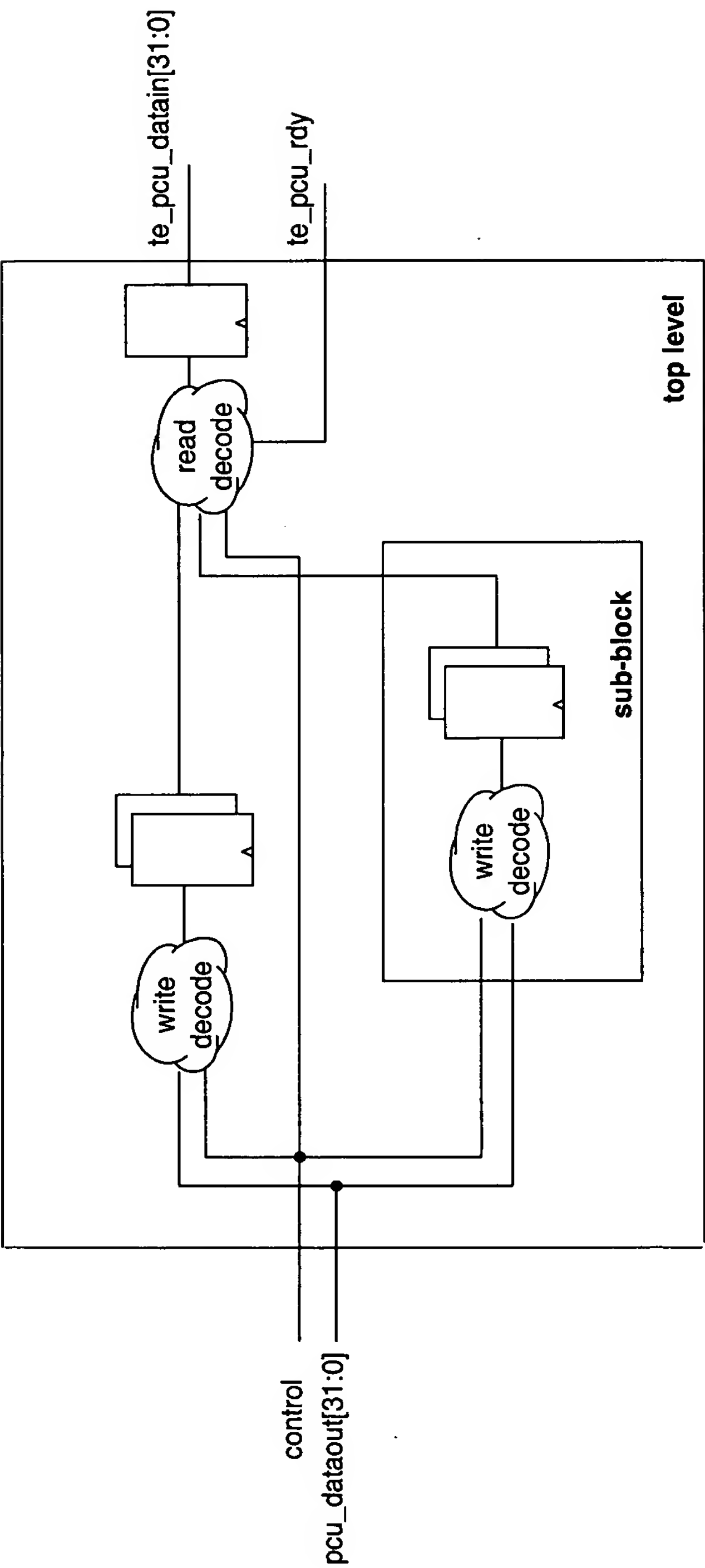


FIG. 189

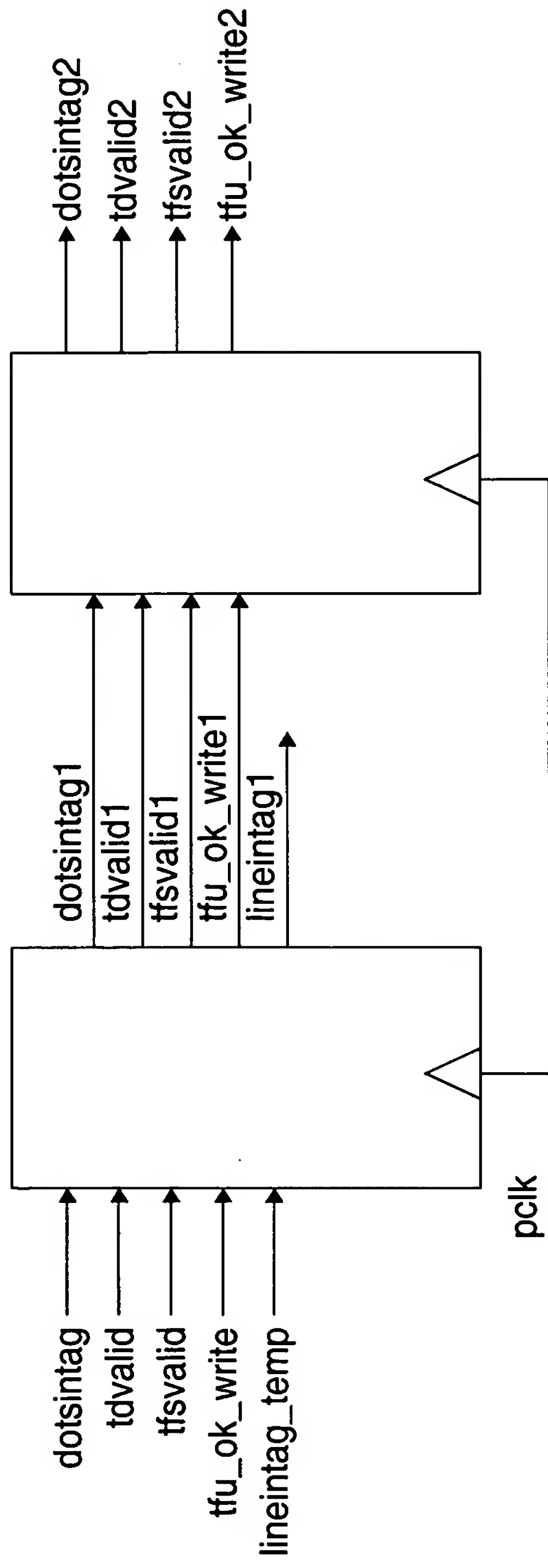


FIG. 191

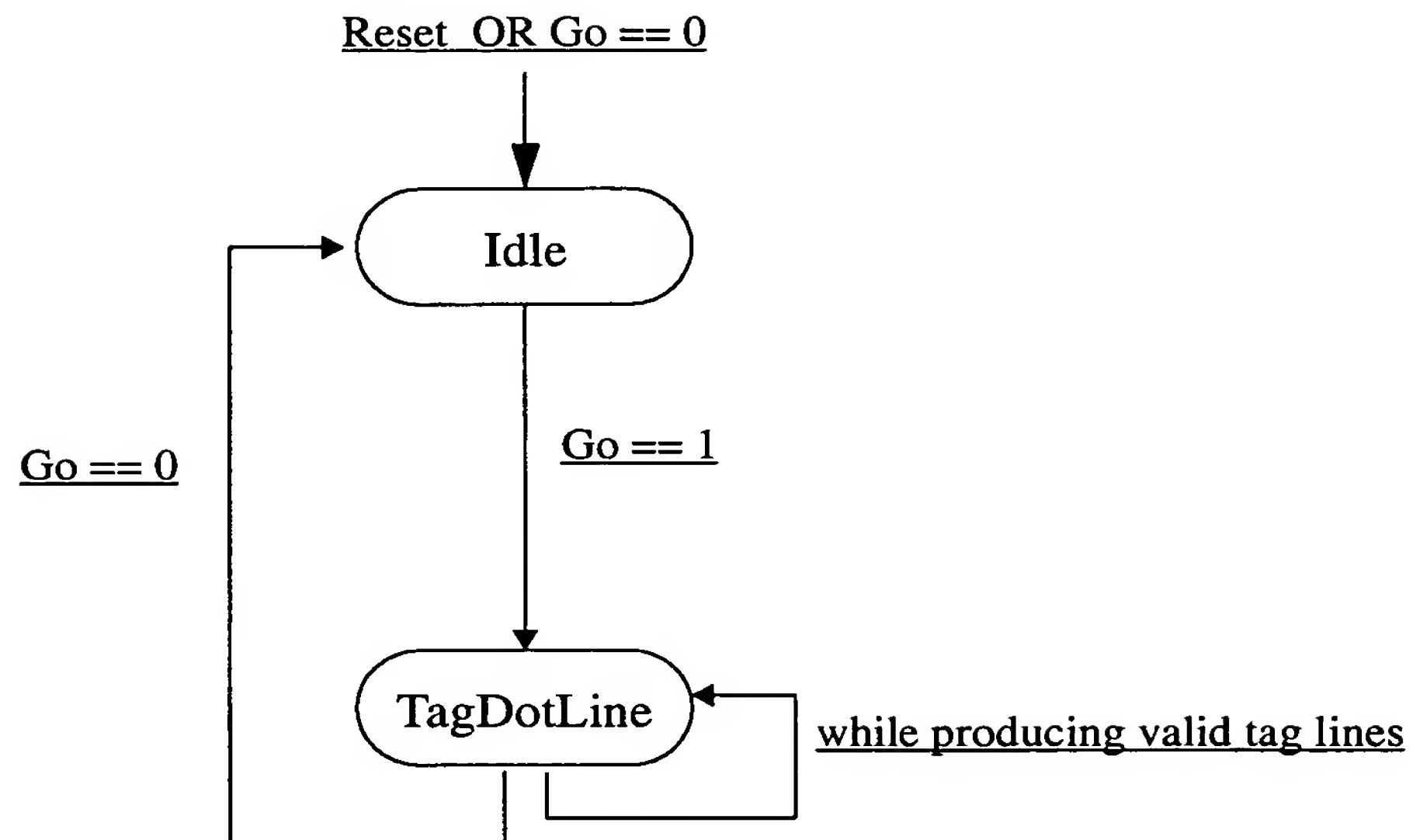


FIG. 190

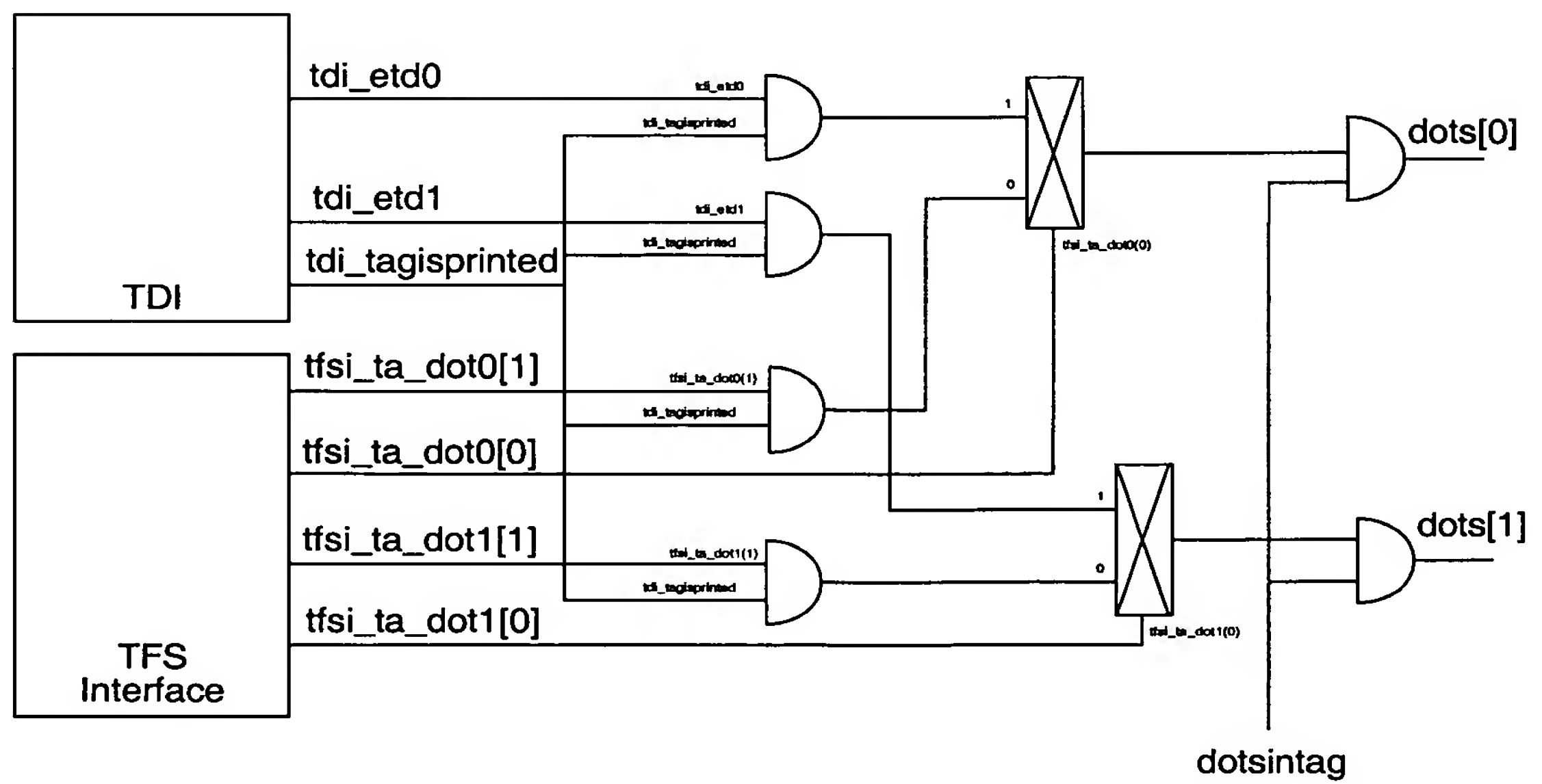


FIG. 192

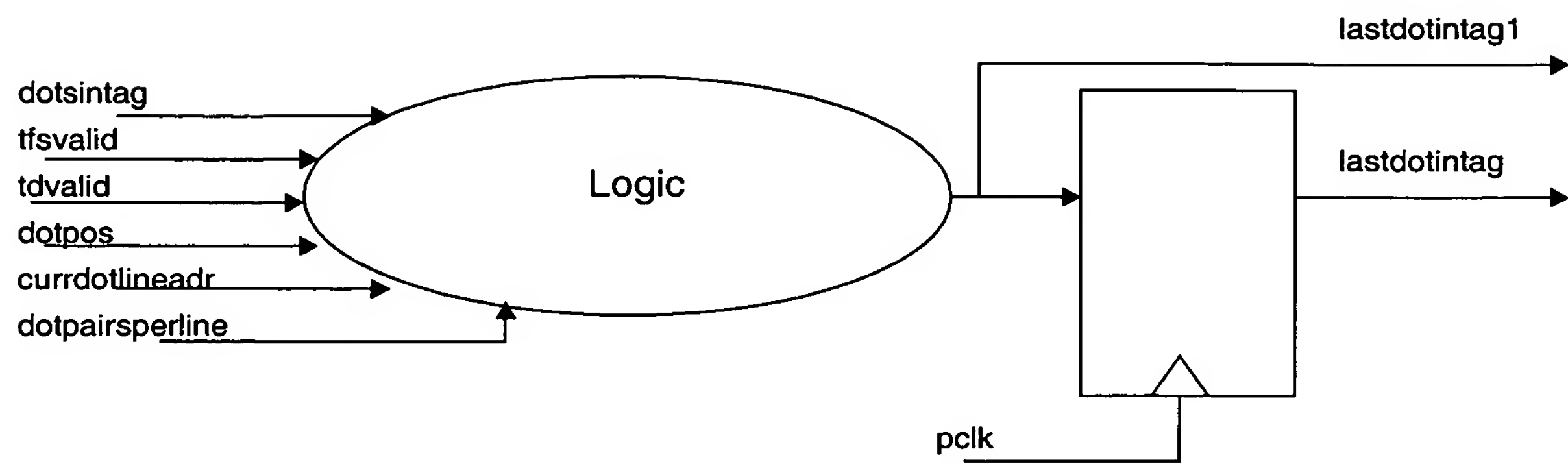


FIG. 193

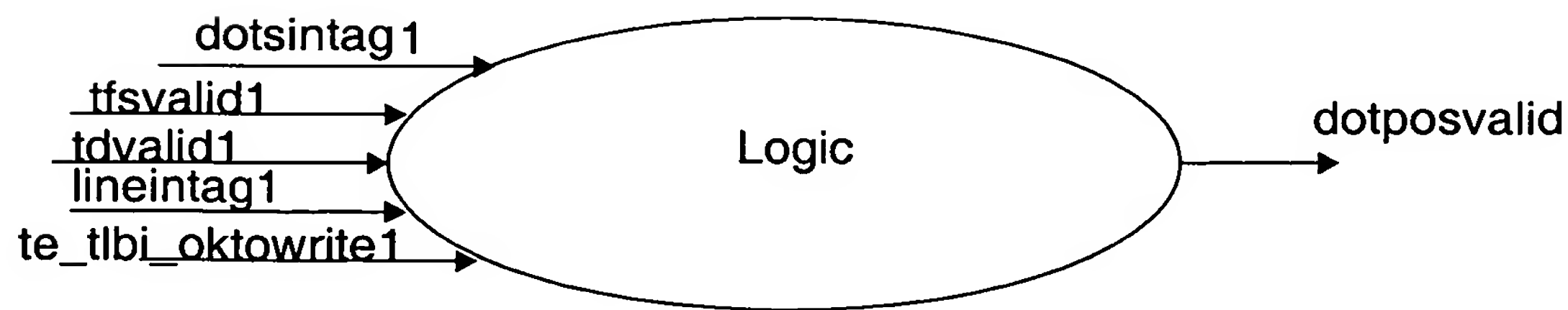


FIG. 194

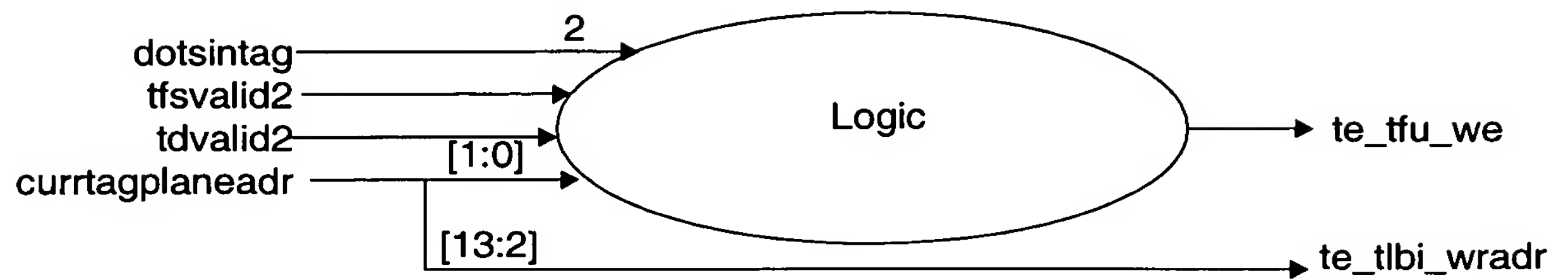


FIG. 195

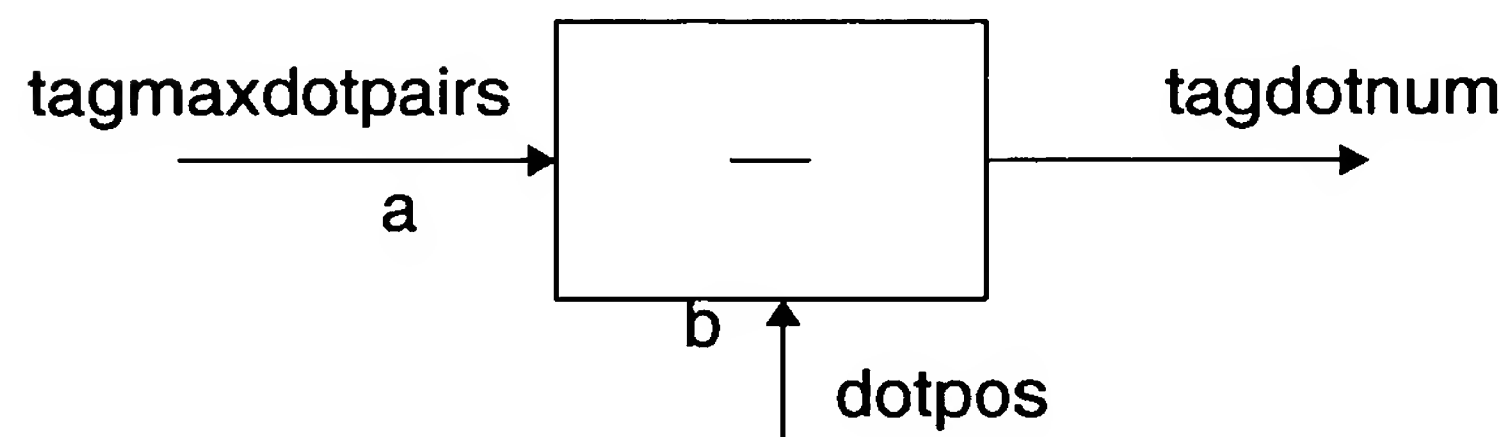


FIG. 196

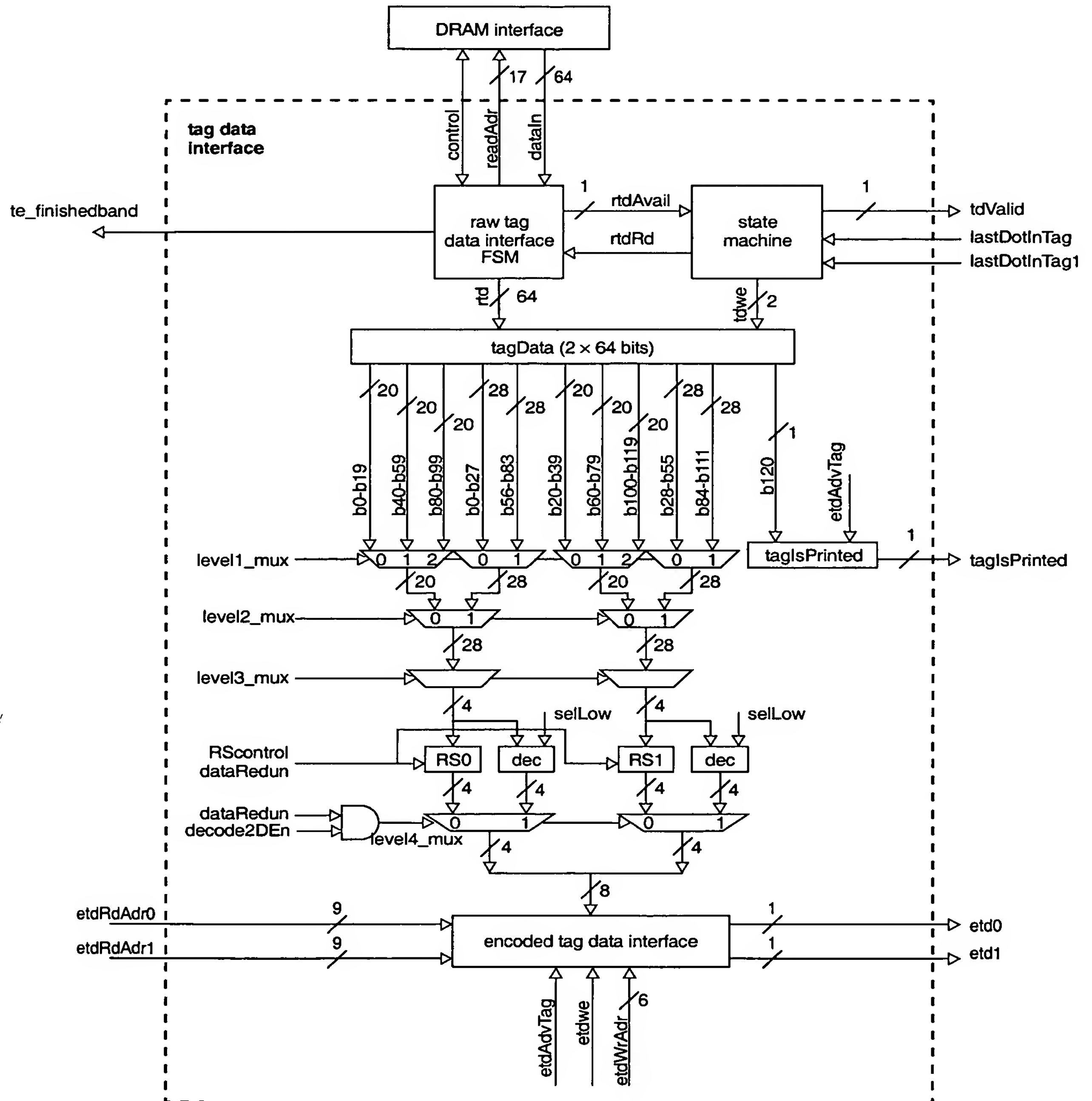
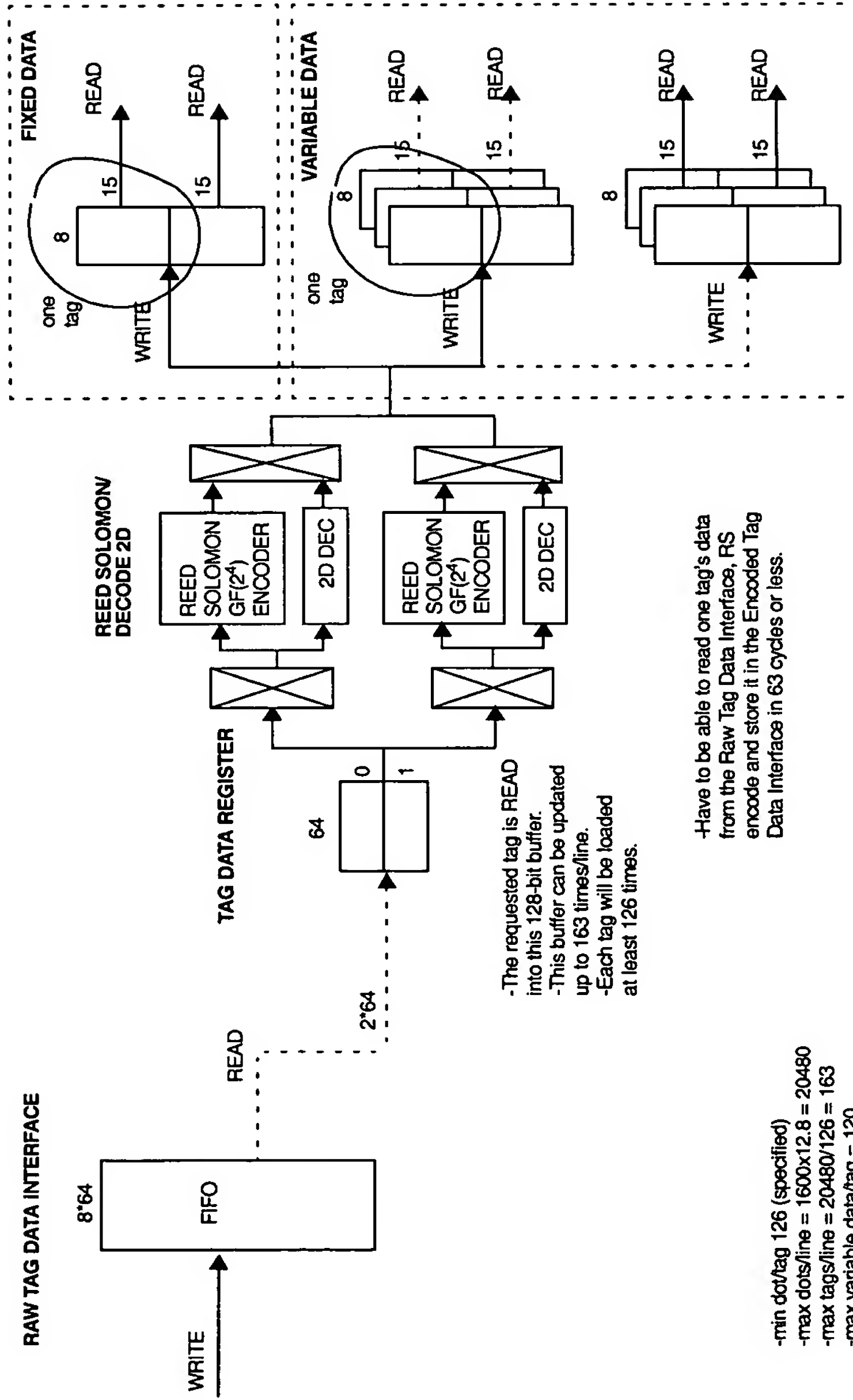


FIG. 197

ENCODED TAG DATA INTERFACE

- Encoded fixed data can be up to 120 bits long
- Use 2 buffers to allow for 2 simultaneously READs in one cycle.
- These stores hold the fixed tag data for 1 tag.
- Total memory = $120 \times 2 = 240$ bits



- min dot/tag 126 (specified)
- max dots/line = $1600 \times 12.8 = 20480$
- max tags/line = $20480 / 126 = 163$
- max variable data/tag = 120
- max amount of tag data/line = 120×164
- Split the 120 tag data bits into 2×64 -bits (8 spare bits)
- Max memory needed for 1 line of tag data = $2 \times 64 \times 164 = 655 \times 32$
- Divide this in half to allow for simultaneous READ/WRITE
- Once all this data is loaded it will be valid for at least 126 lines.
- From the specification, we must be able to process 2 dots/cycle.
- 126 lines contains $20480 \times 126 = 2590480$ dots.
- Therefore the data will be updated at most every 1290240 cycles.
- Total memory = $164 \times 2 \times 64 = 20992$ -bits
- The store uses 9-bit addressing. Bit-9 indicates which buffer.
- Once printing has started each half buffer has 1/2 a line in which to i.e. for a 12.8 inch line it has 10240 dots or 5120 cycles for an 8 inch line it has 6400 dots or 3200 cycles

FIG. 198

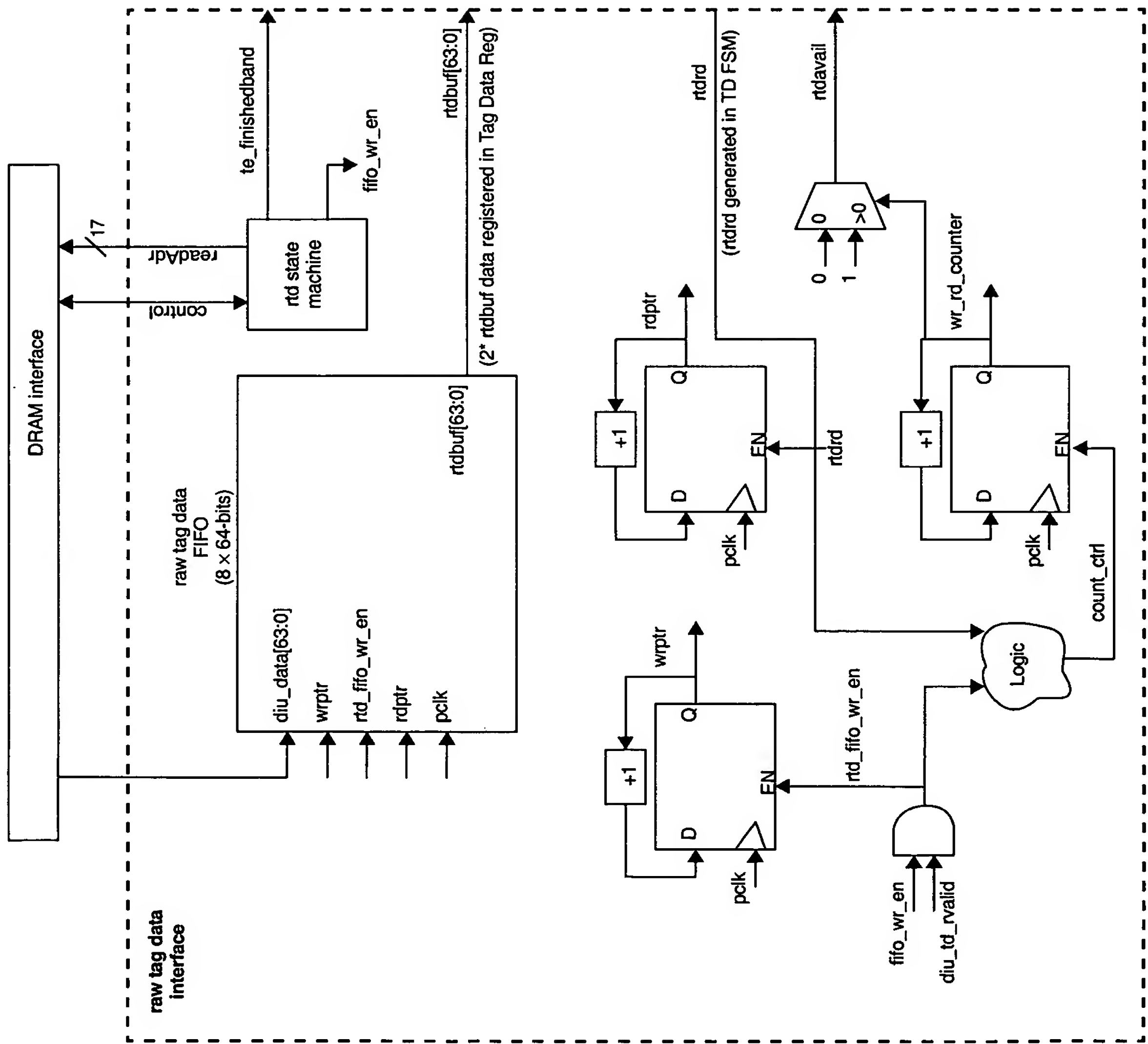


FIG. 199

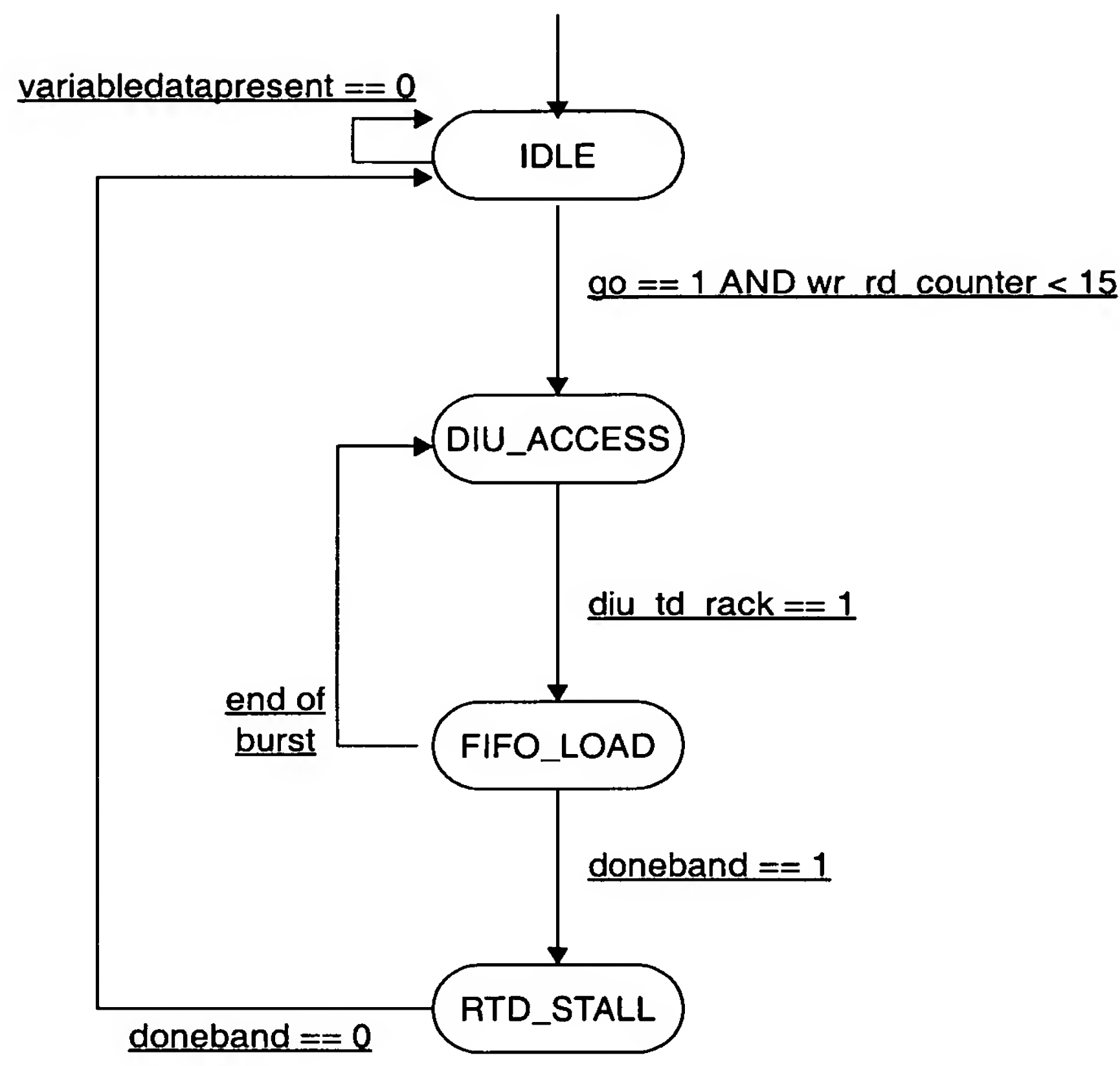


FIG. 200

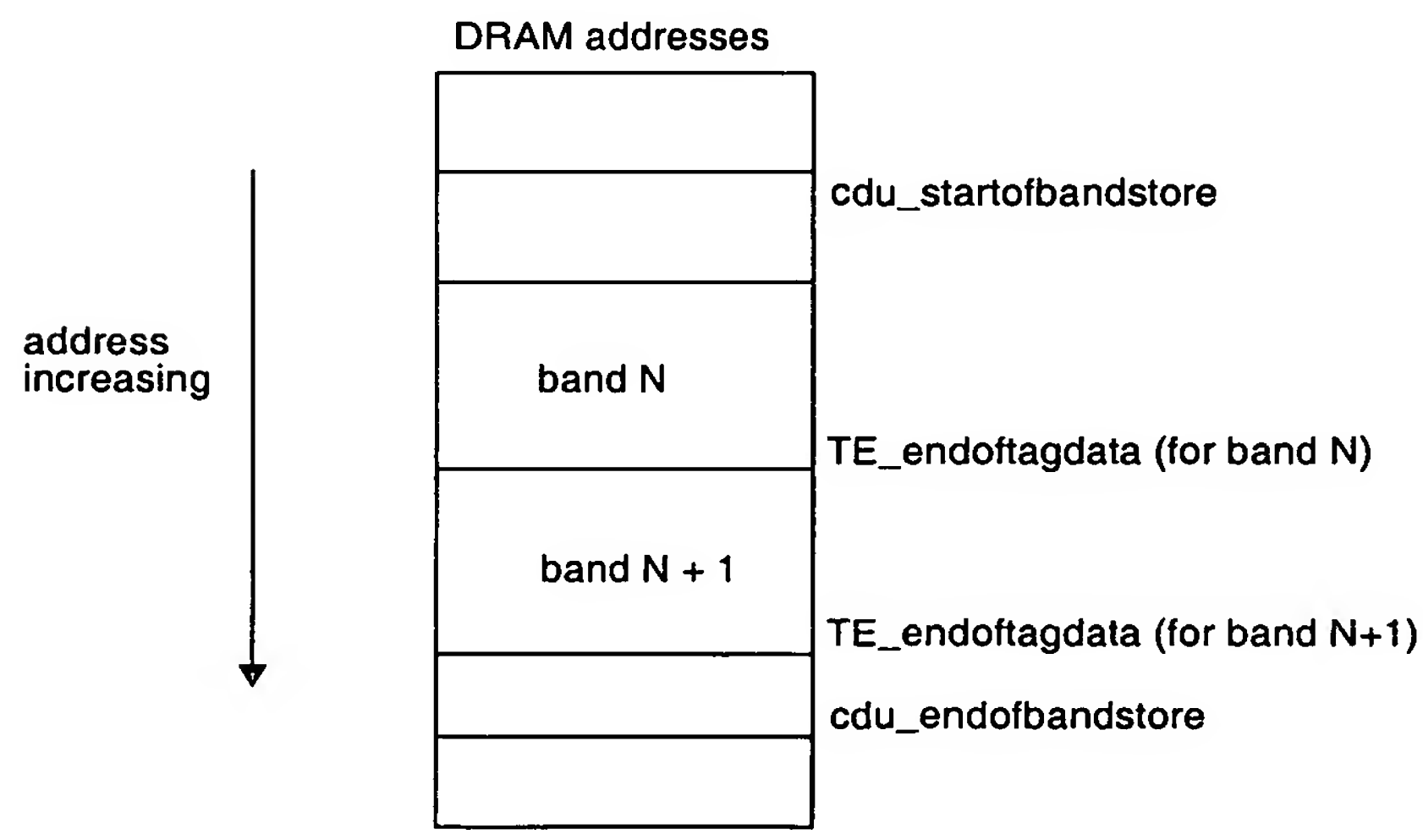


FIG. 201

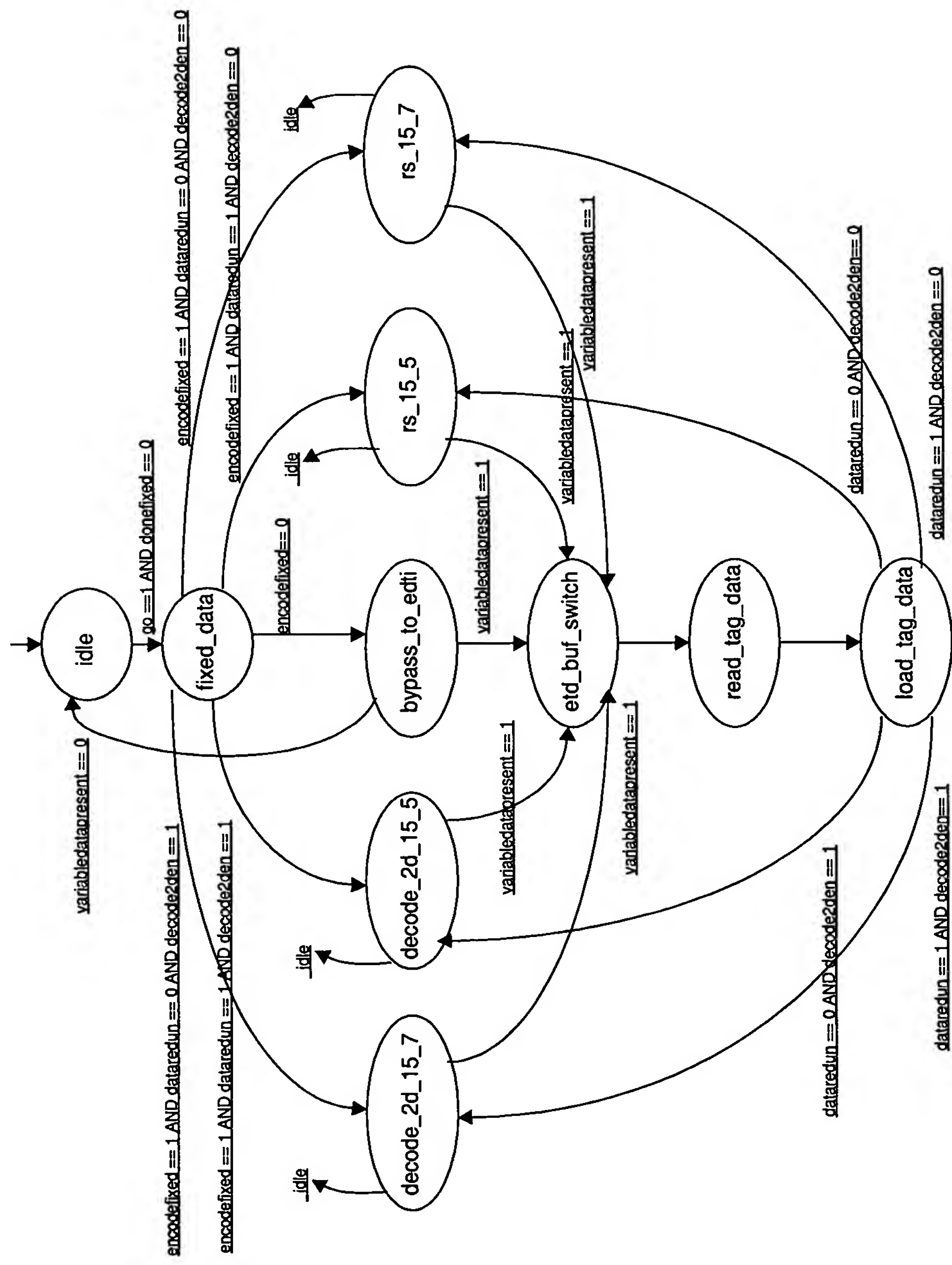


FIG. 202

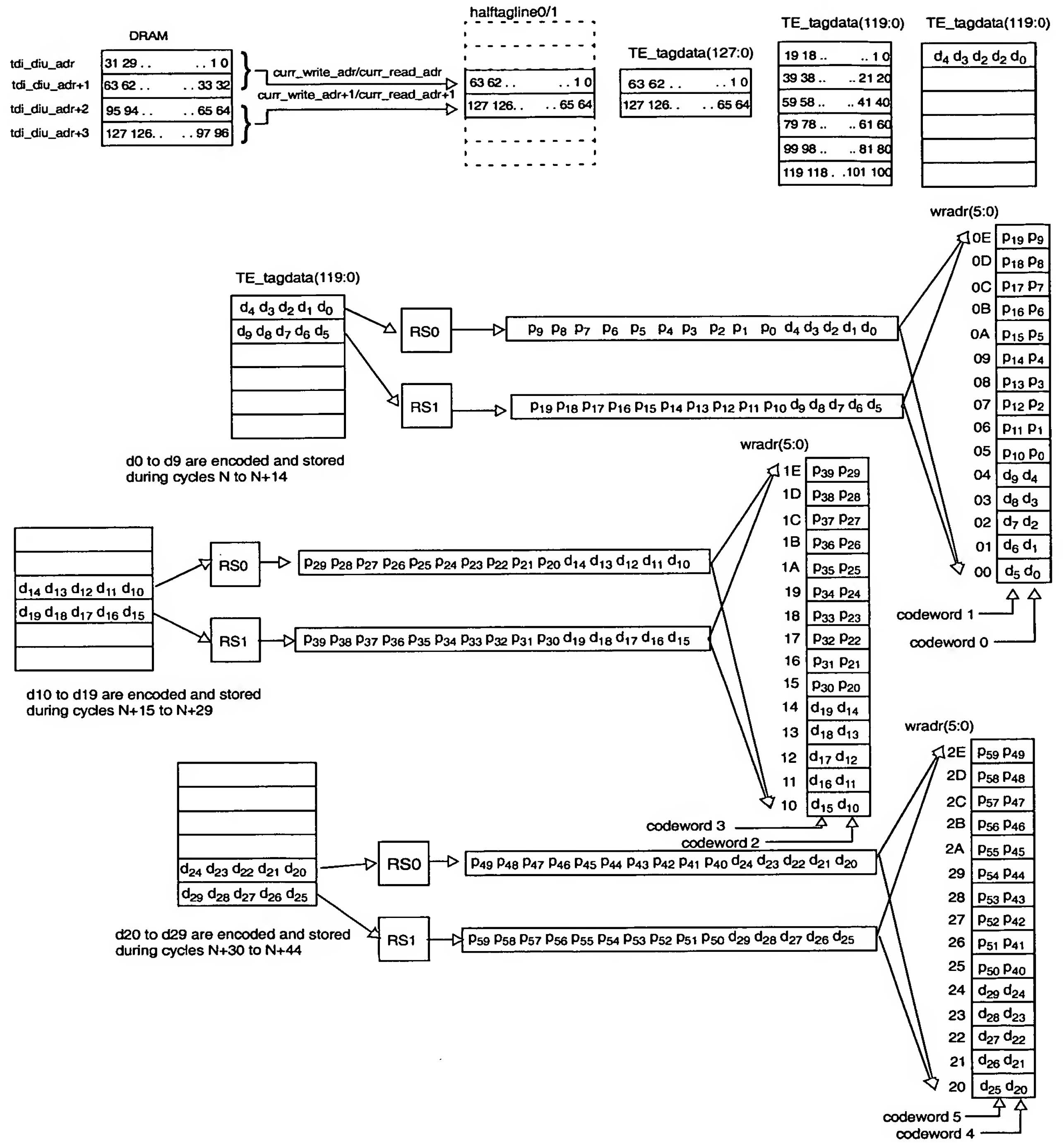


FIG. 203

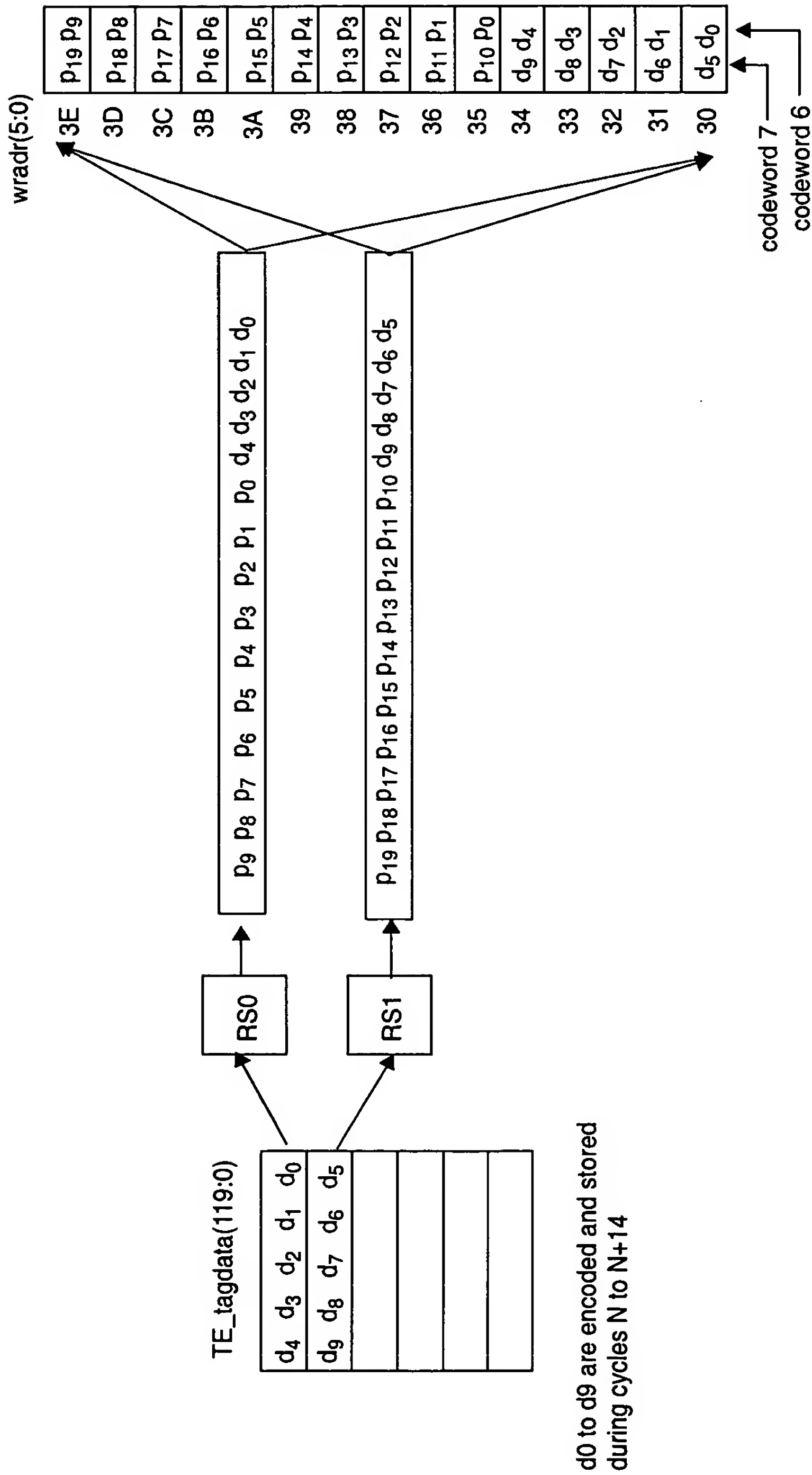


FIG. 204

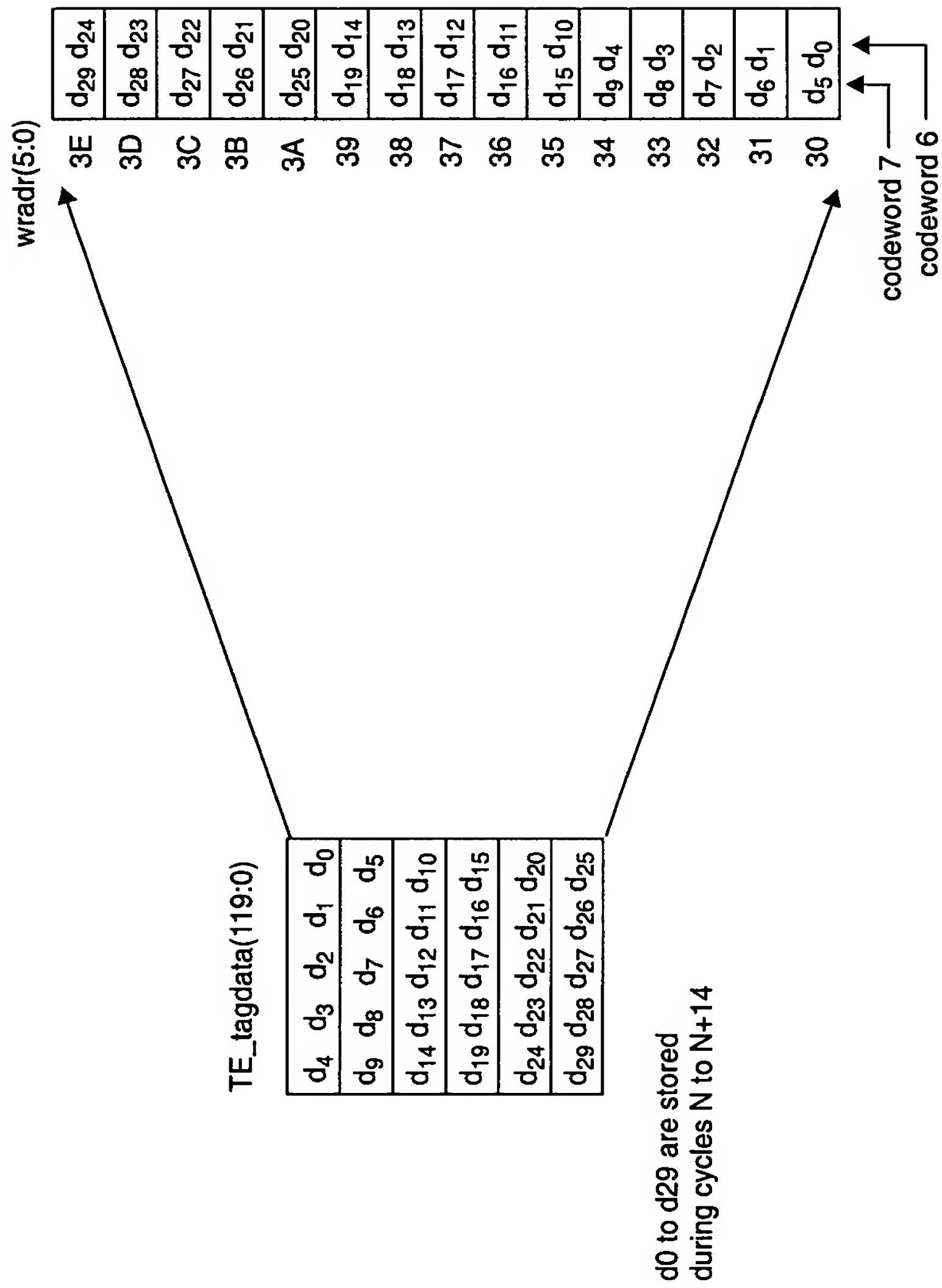


FIG. 205

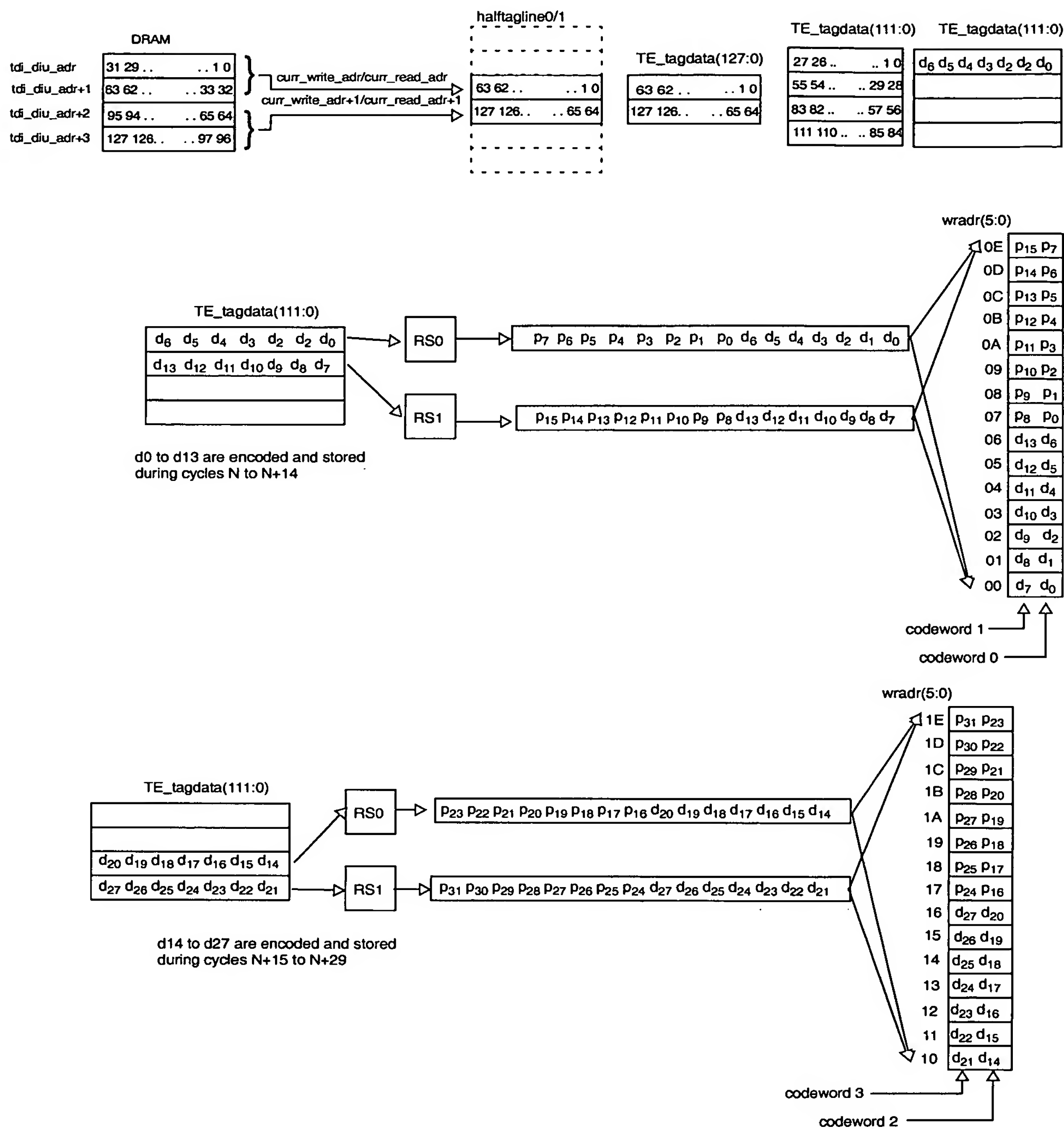


FIG. 206

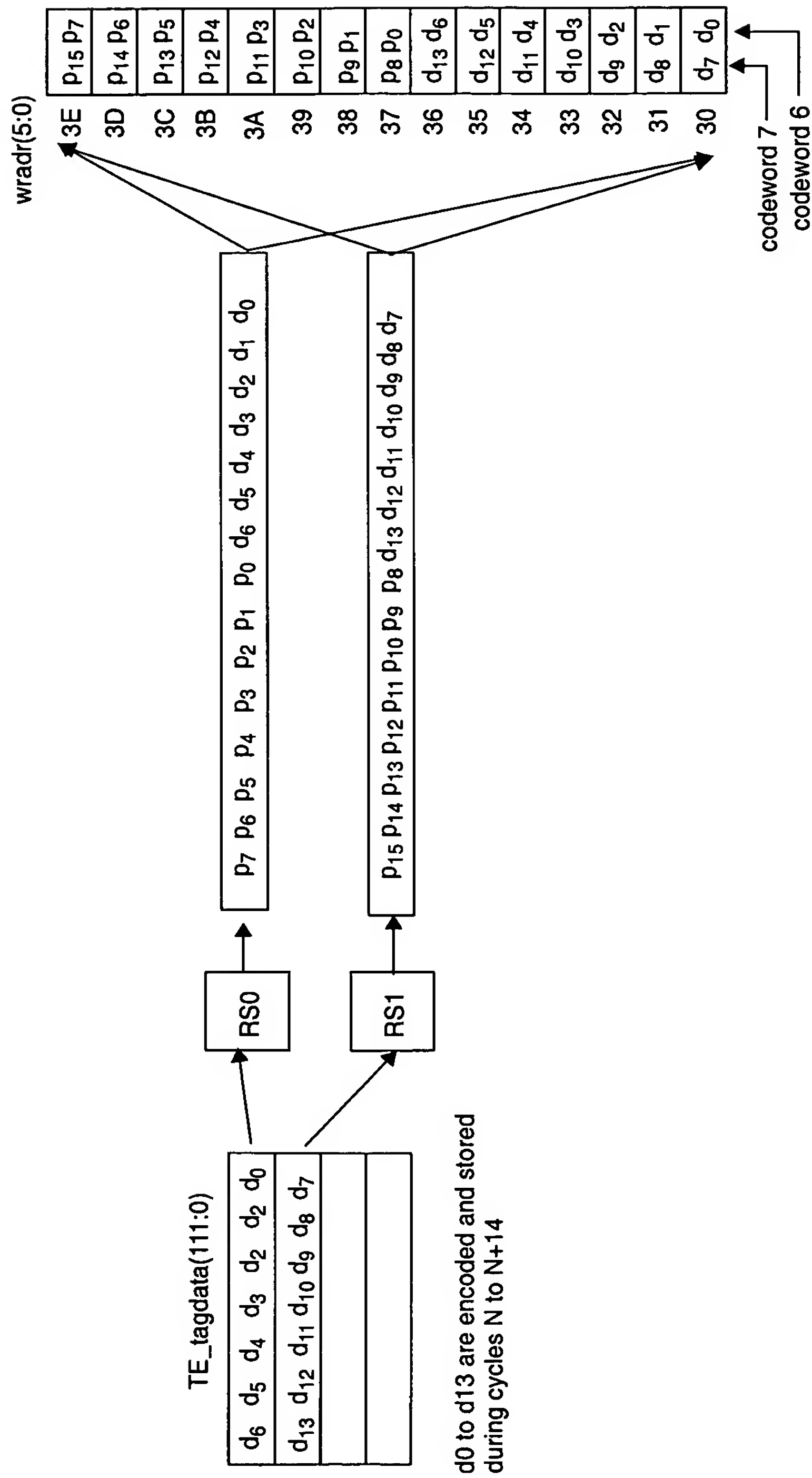


FIG. 207

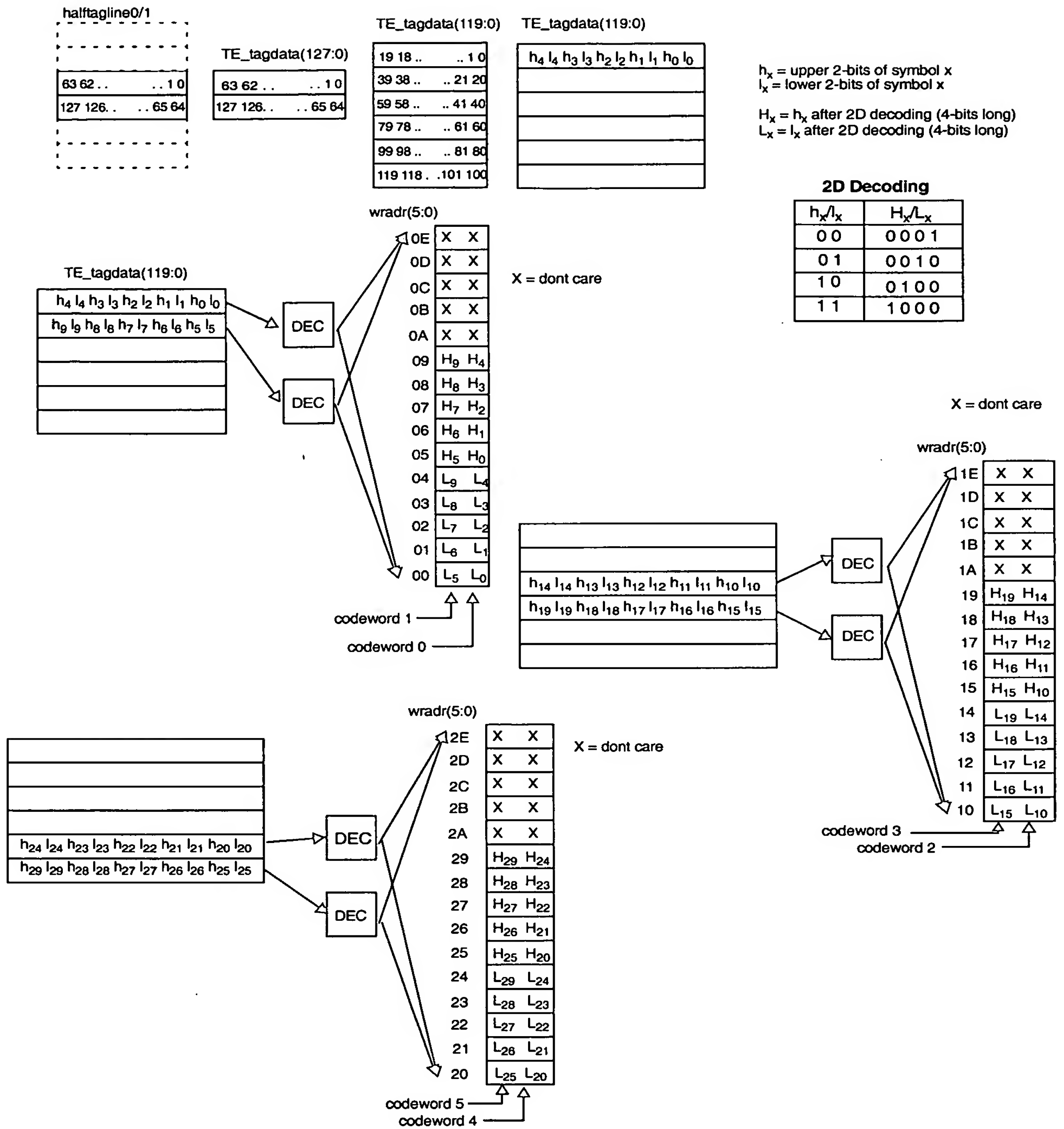


FIG. 208

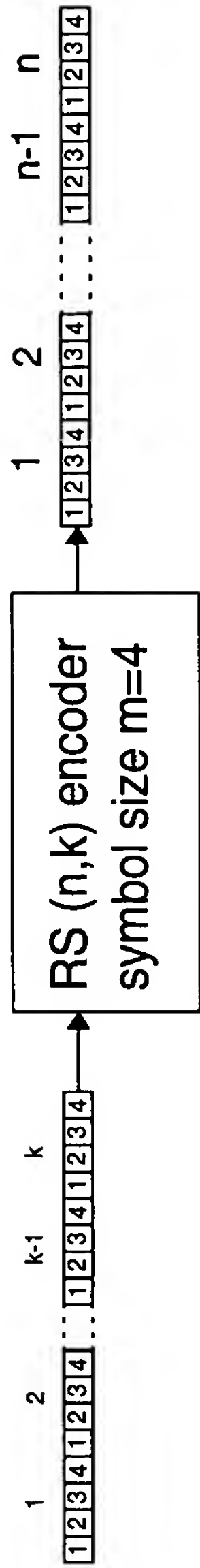


FIG. 209

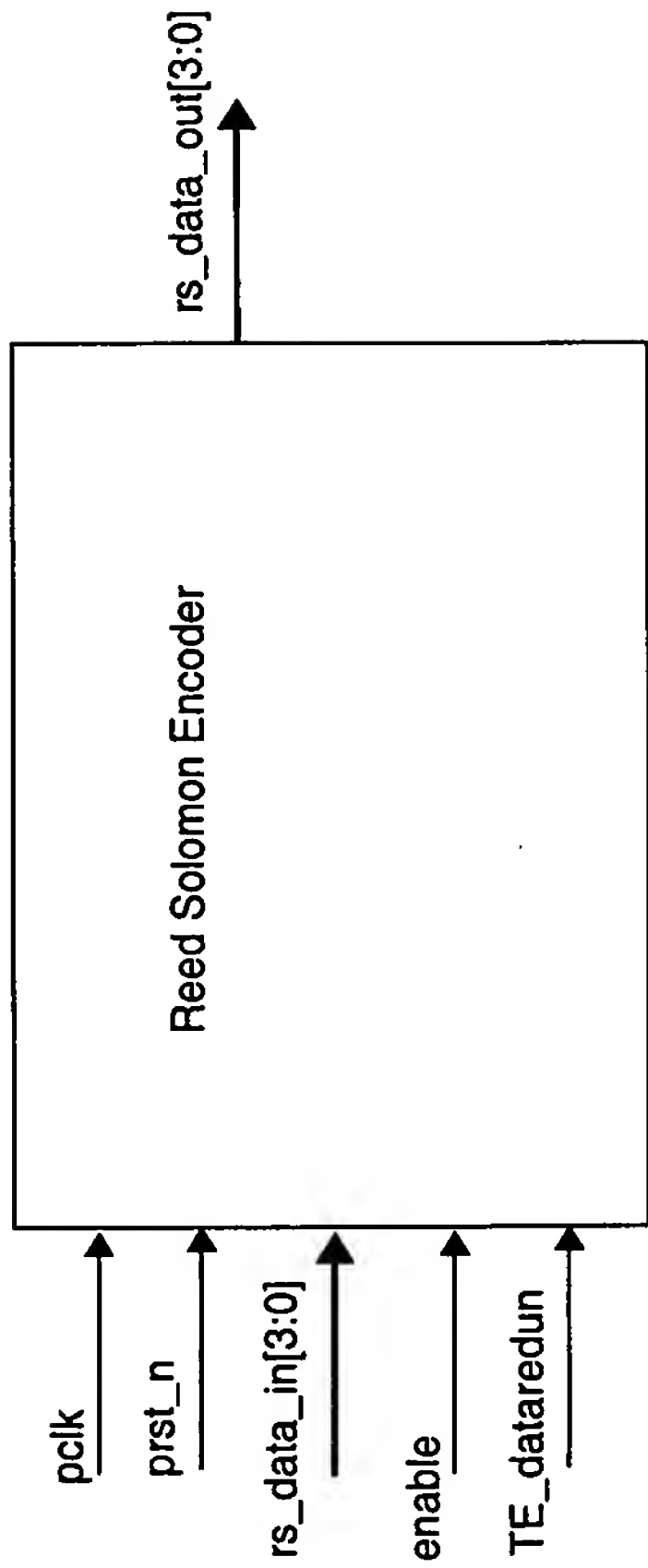


FIG. 210

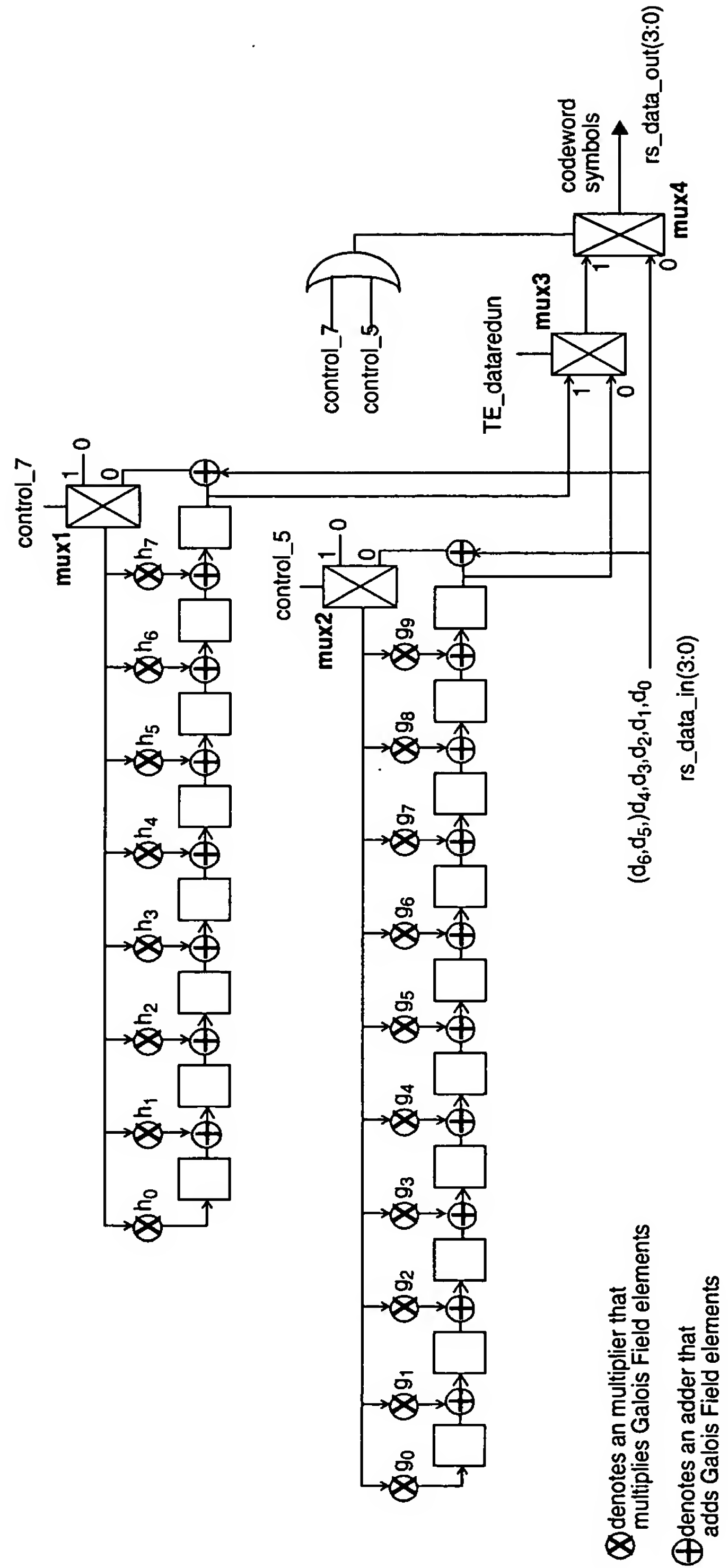


FIG. 211

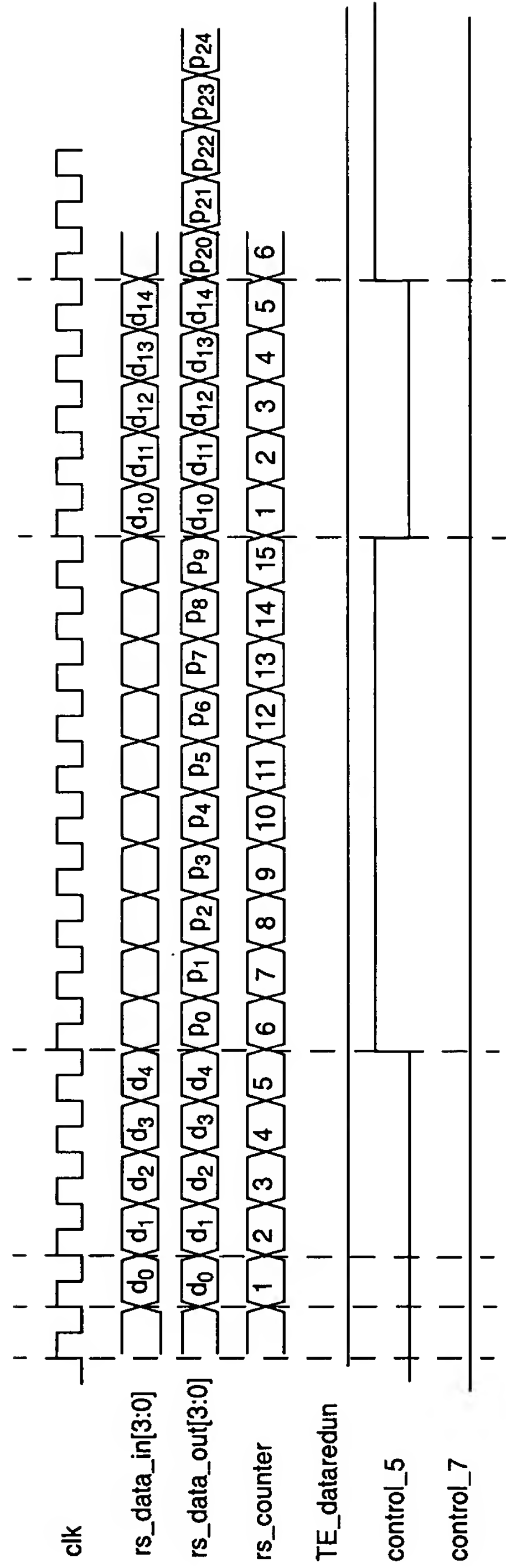


FIG. 212

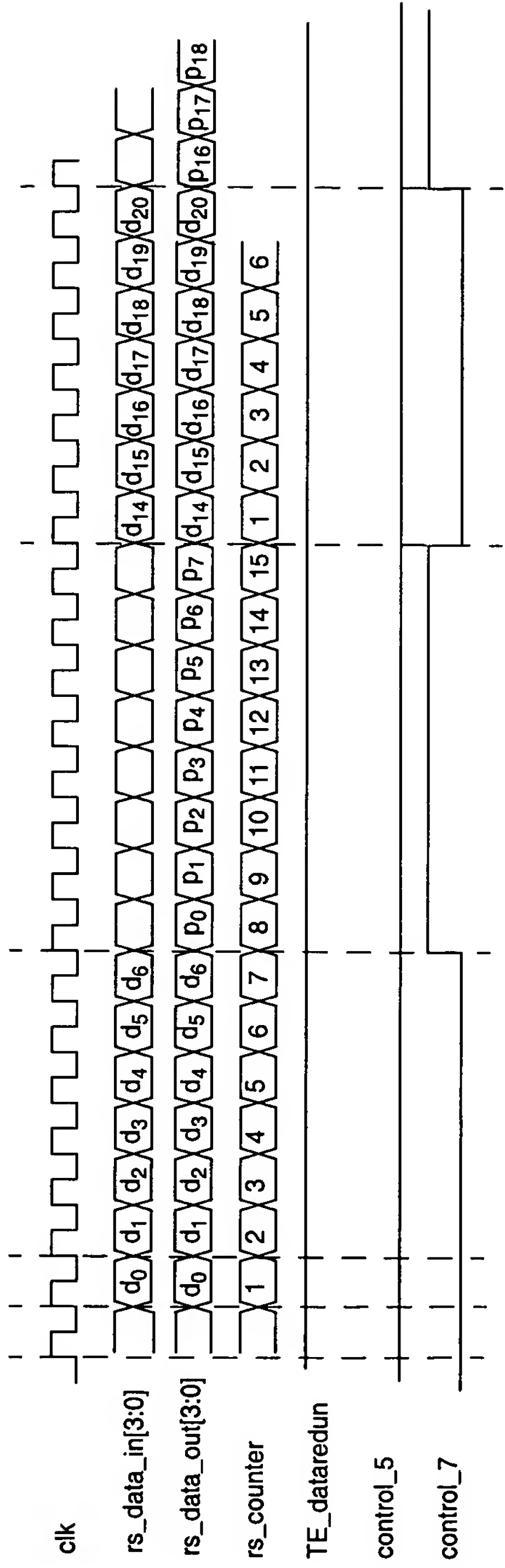


FIG. 213

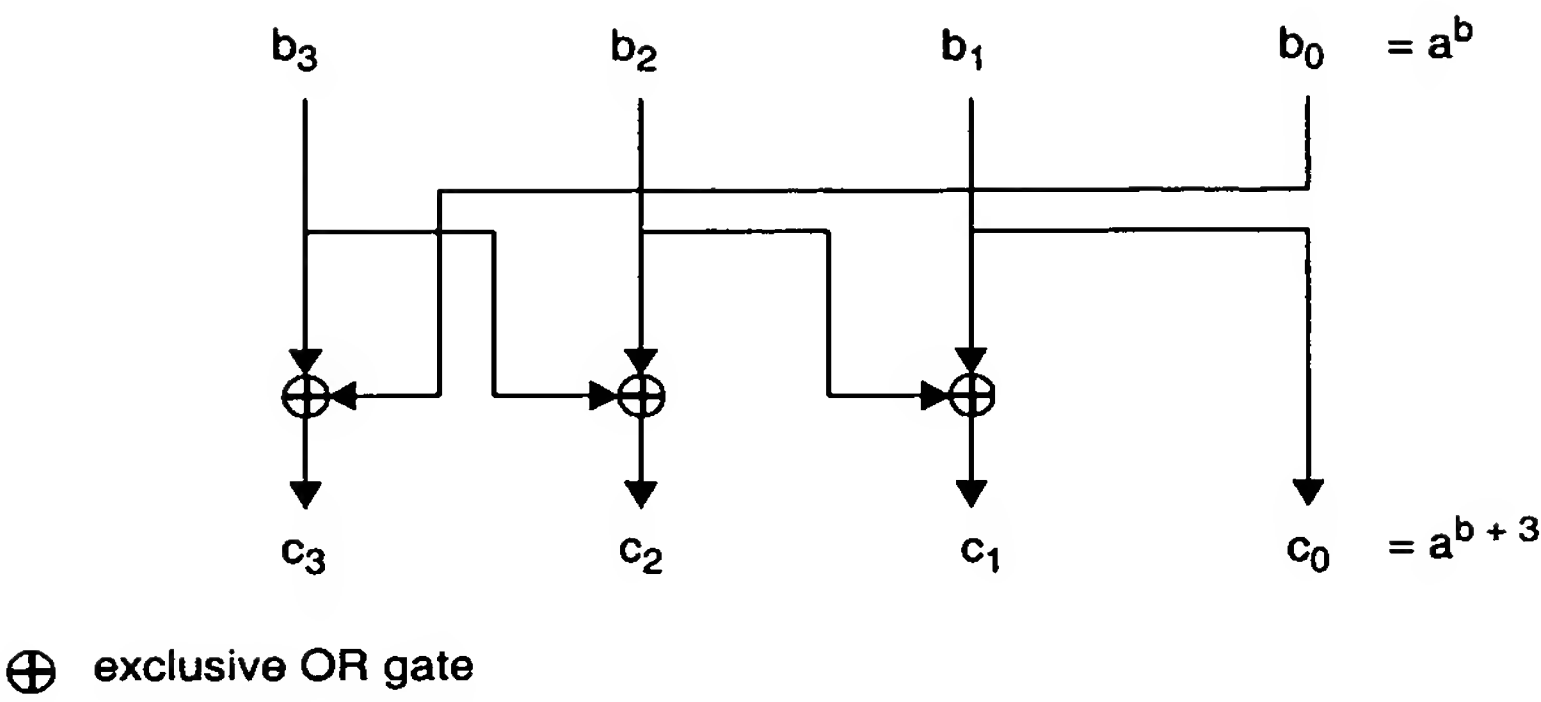


FIG. 214

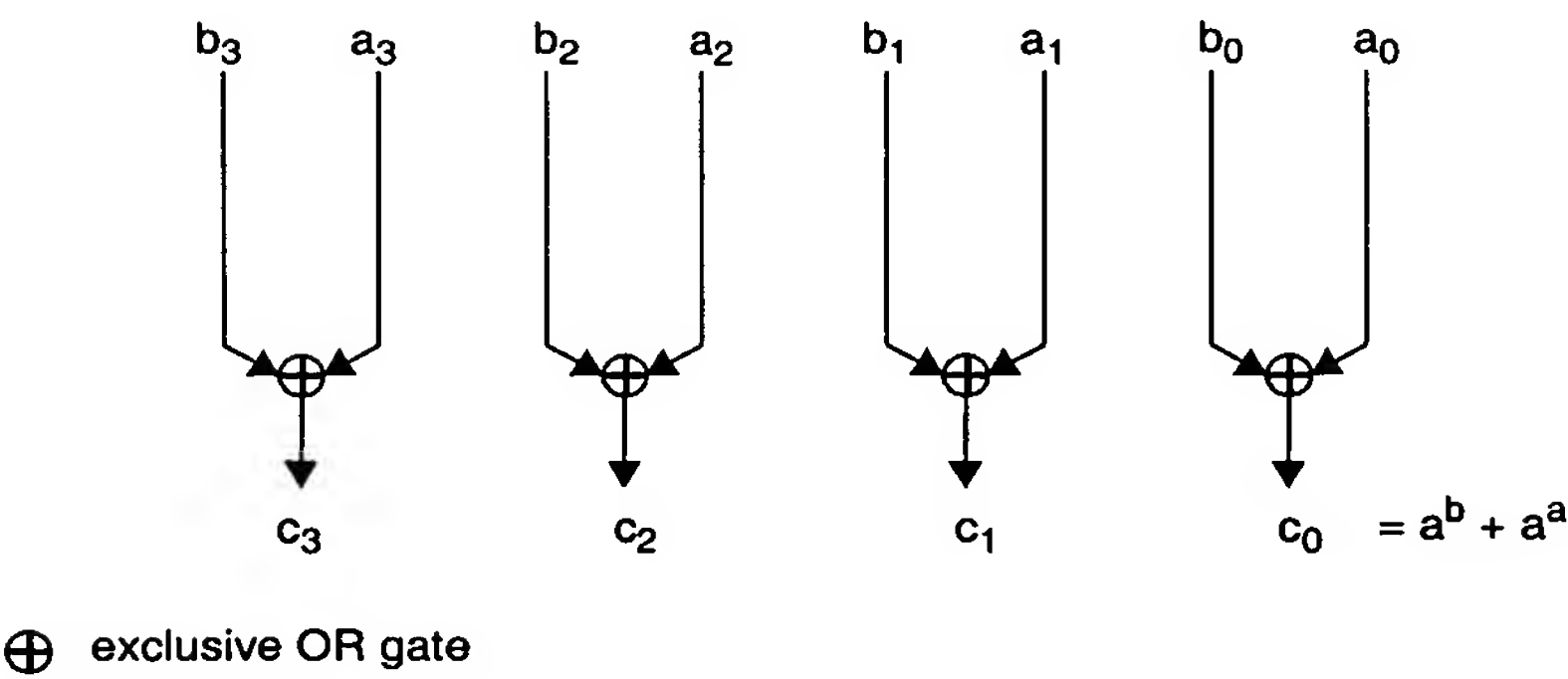


FIG. 215

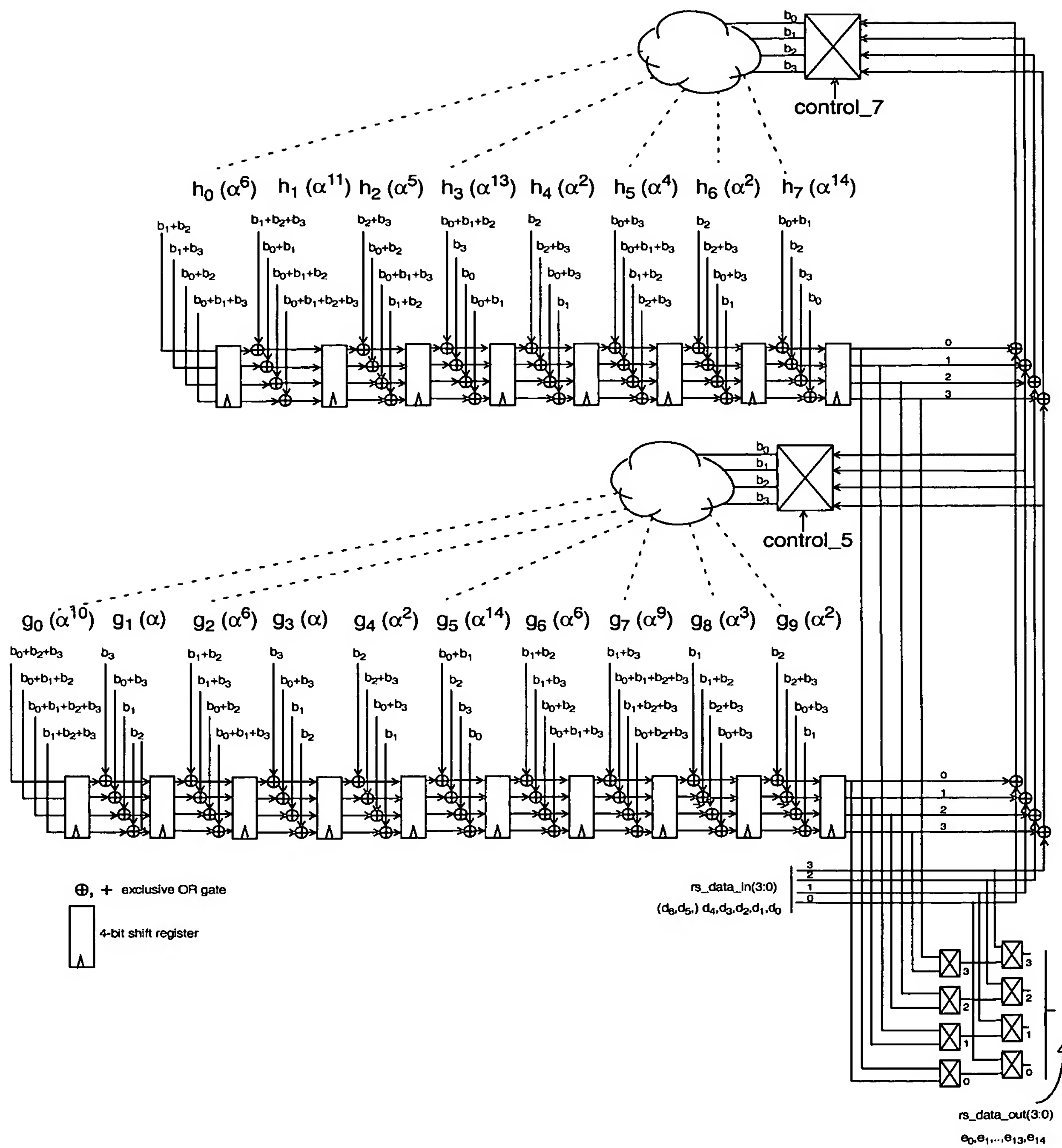


FIG. 216

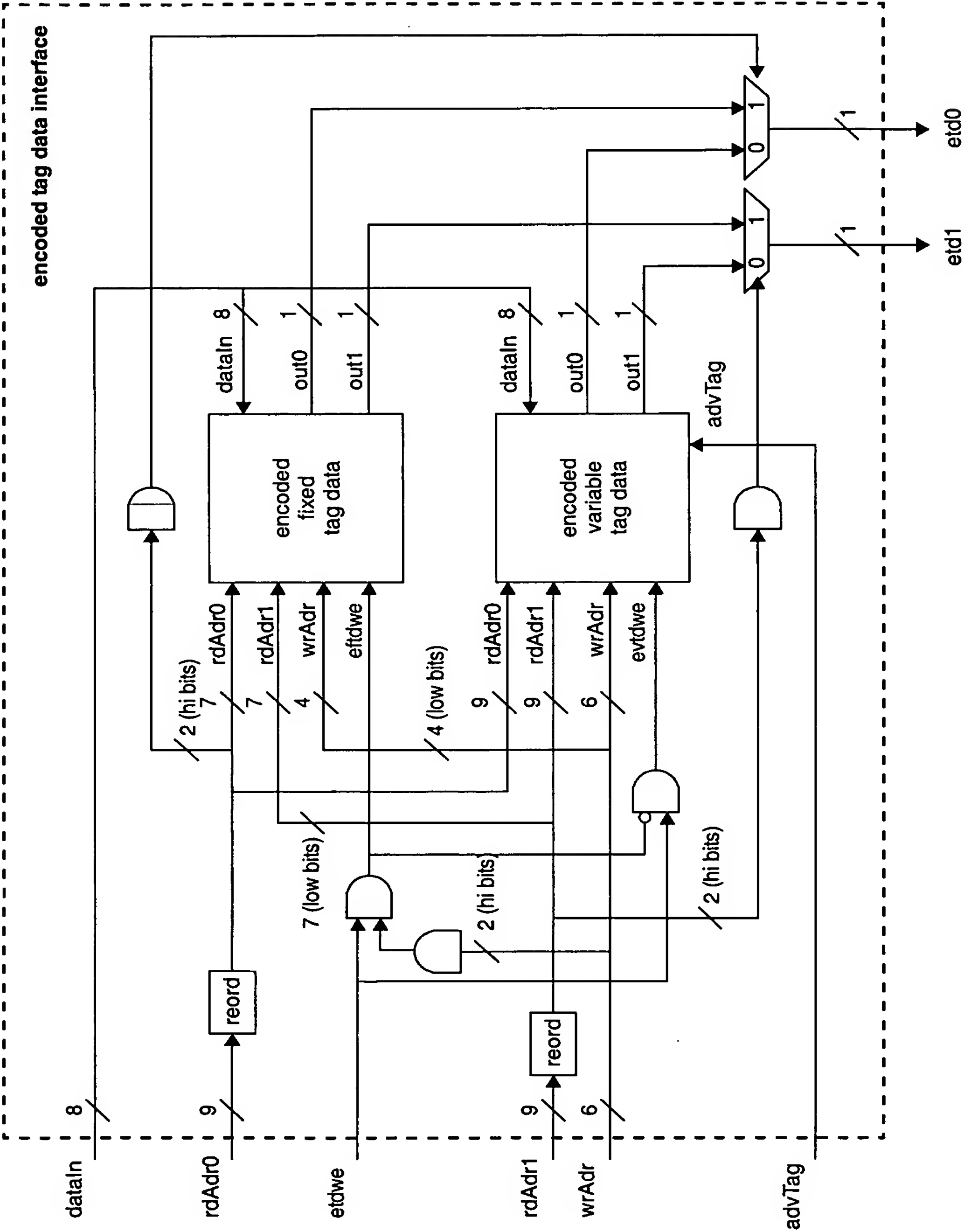


FIG. 217

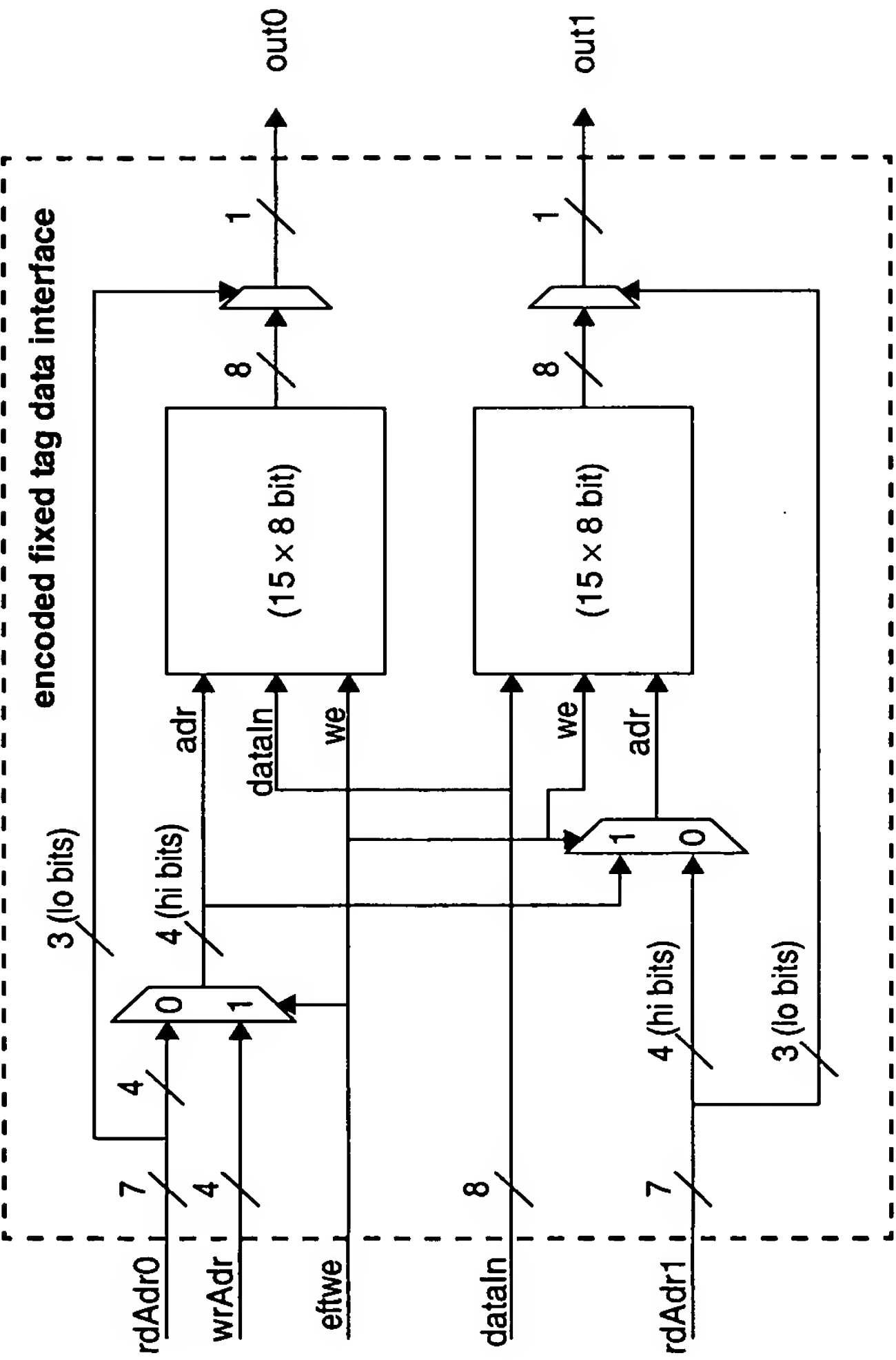


FIG. 218

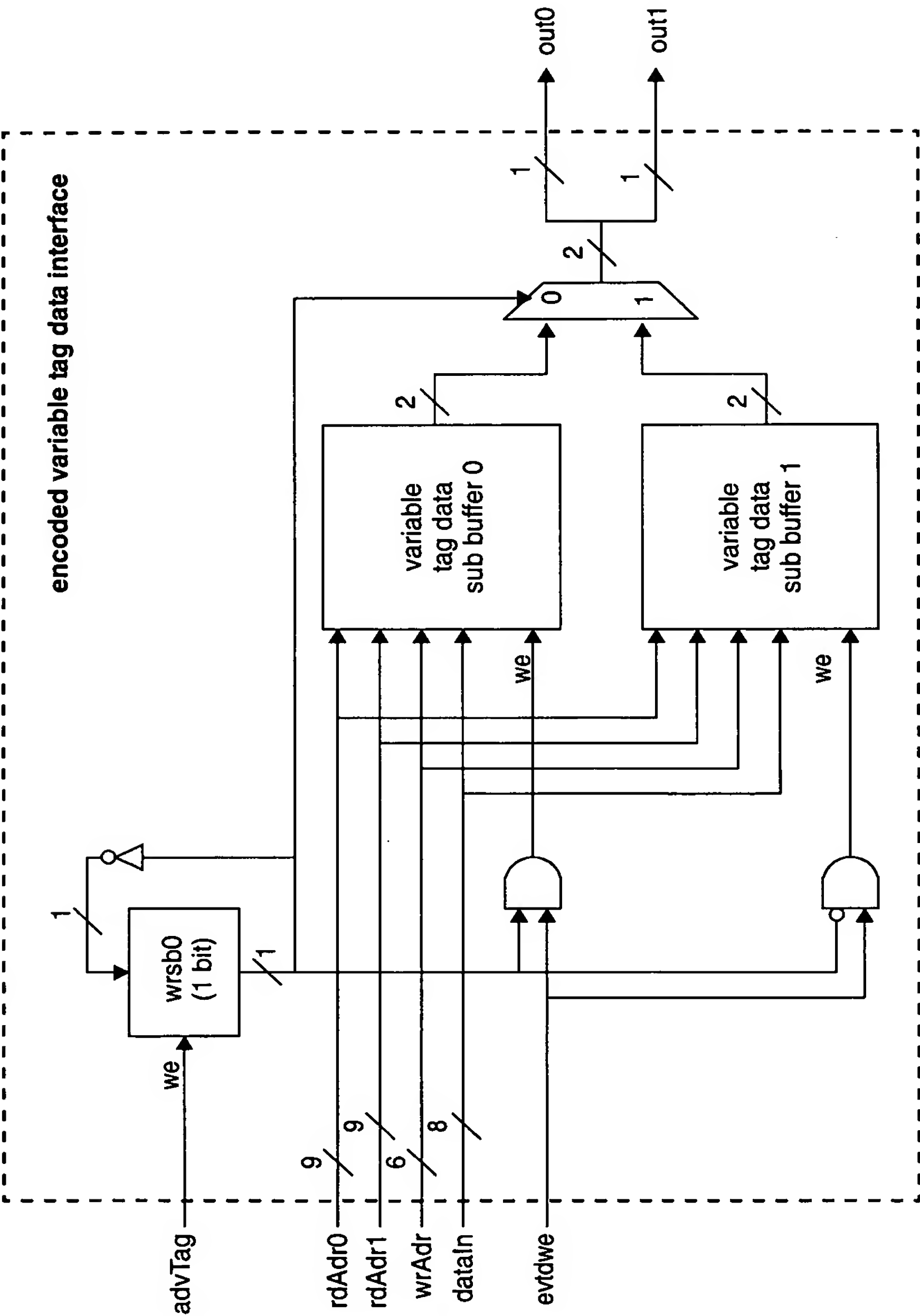


FIG. 219

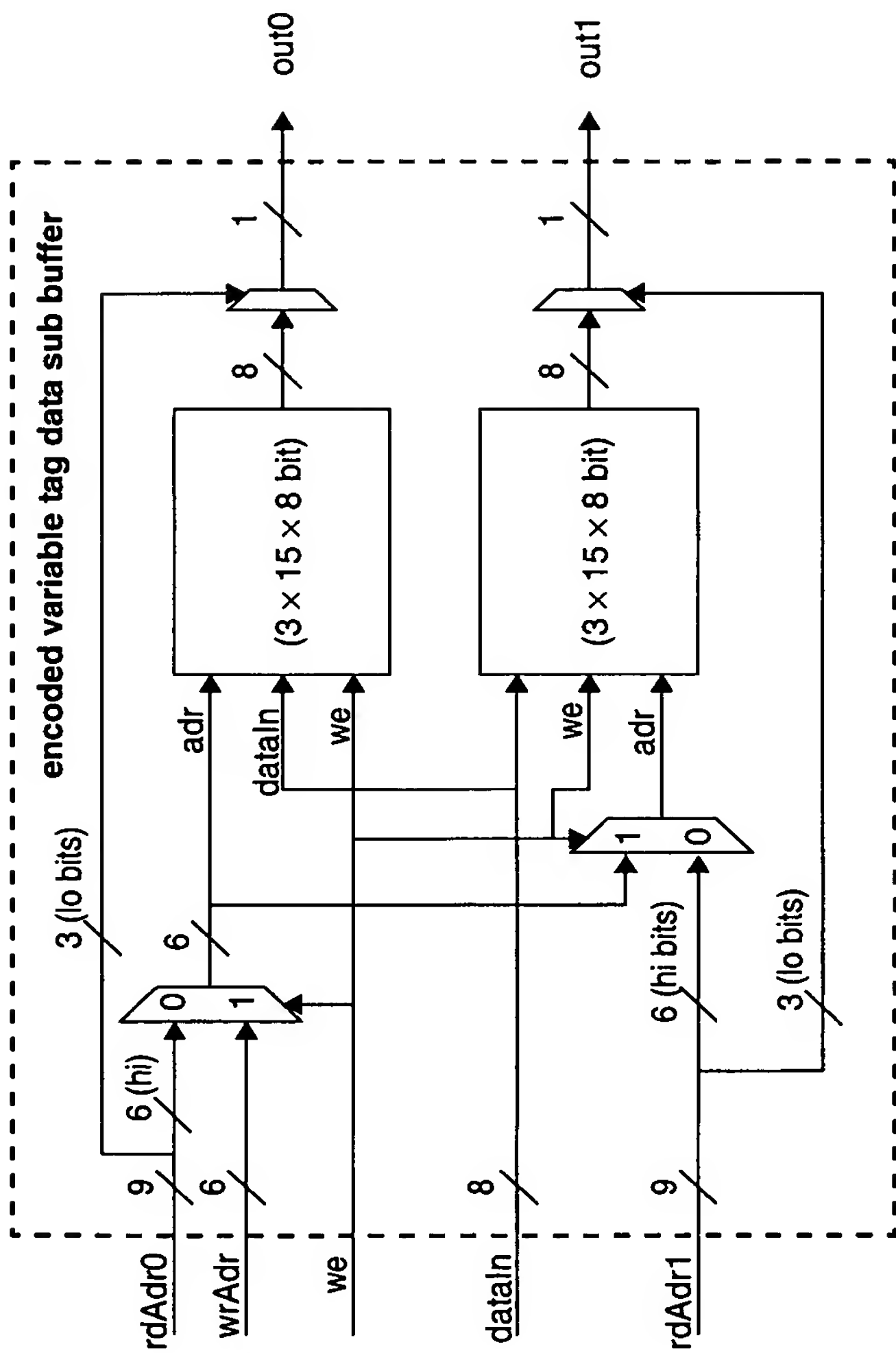


FIG. 220

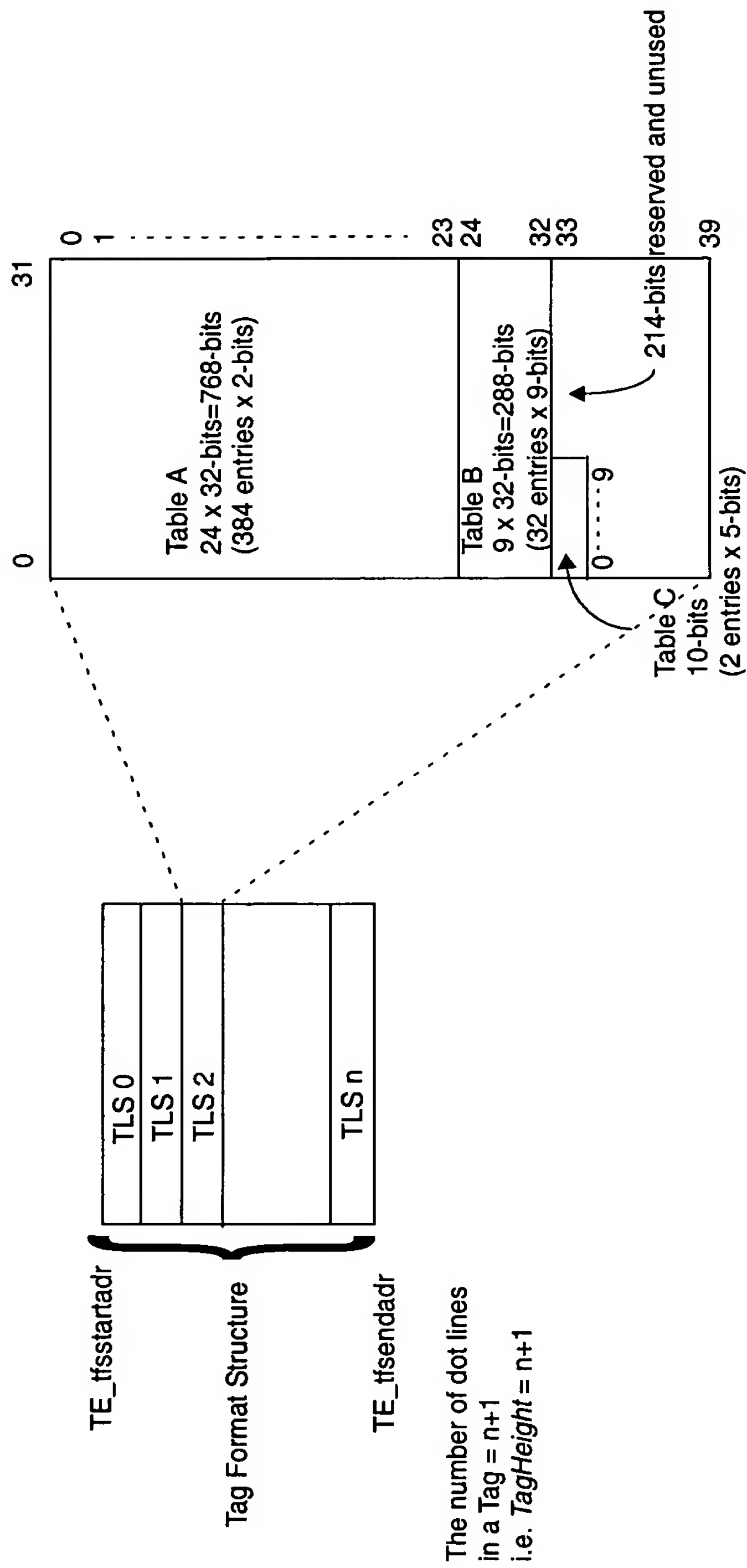


FIG. 221

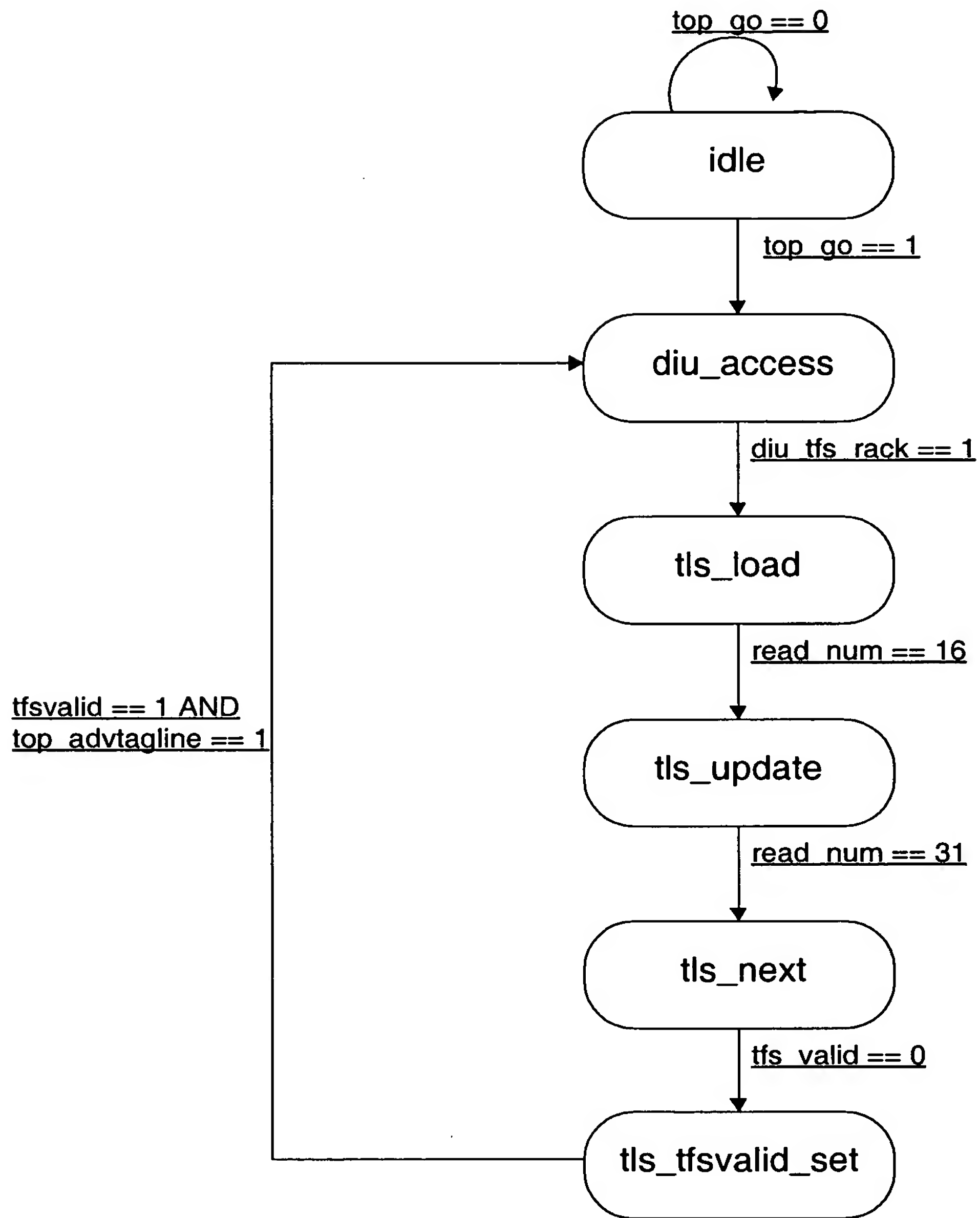


FIG. 222

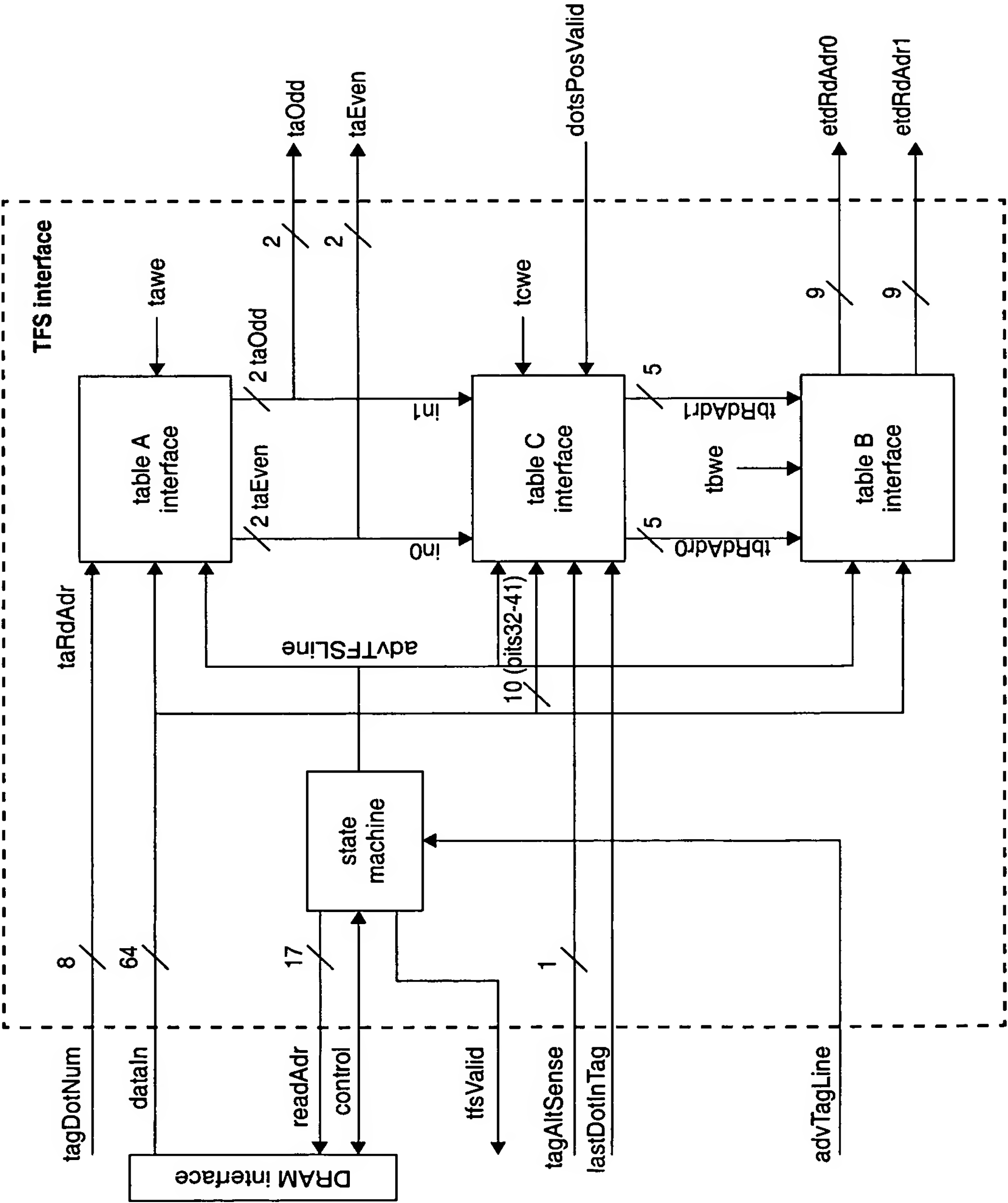


FIG. 223

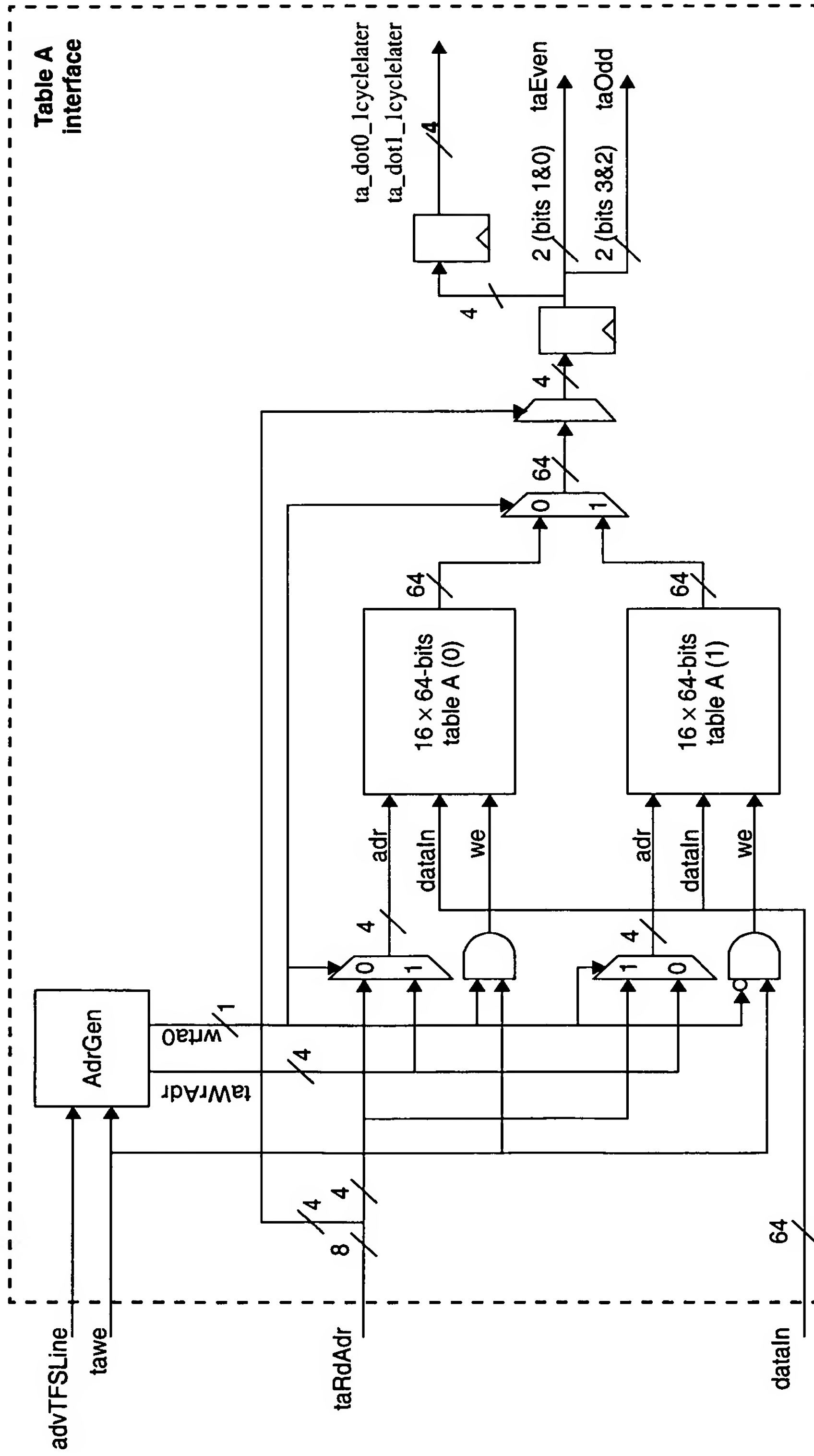


FIG. 224

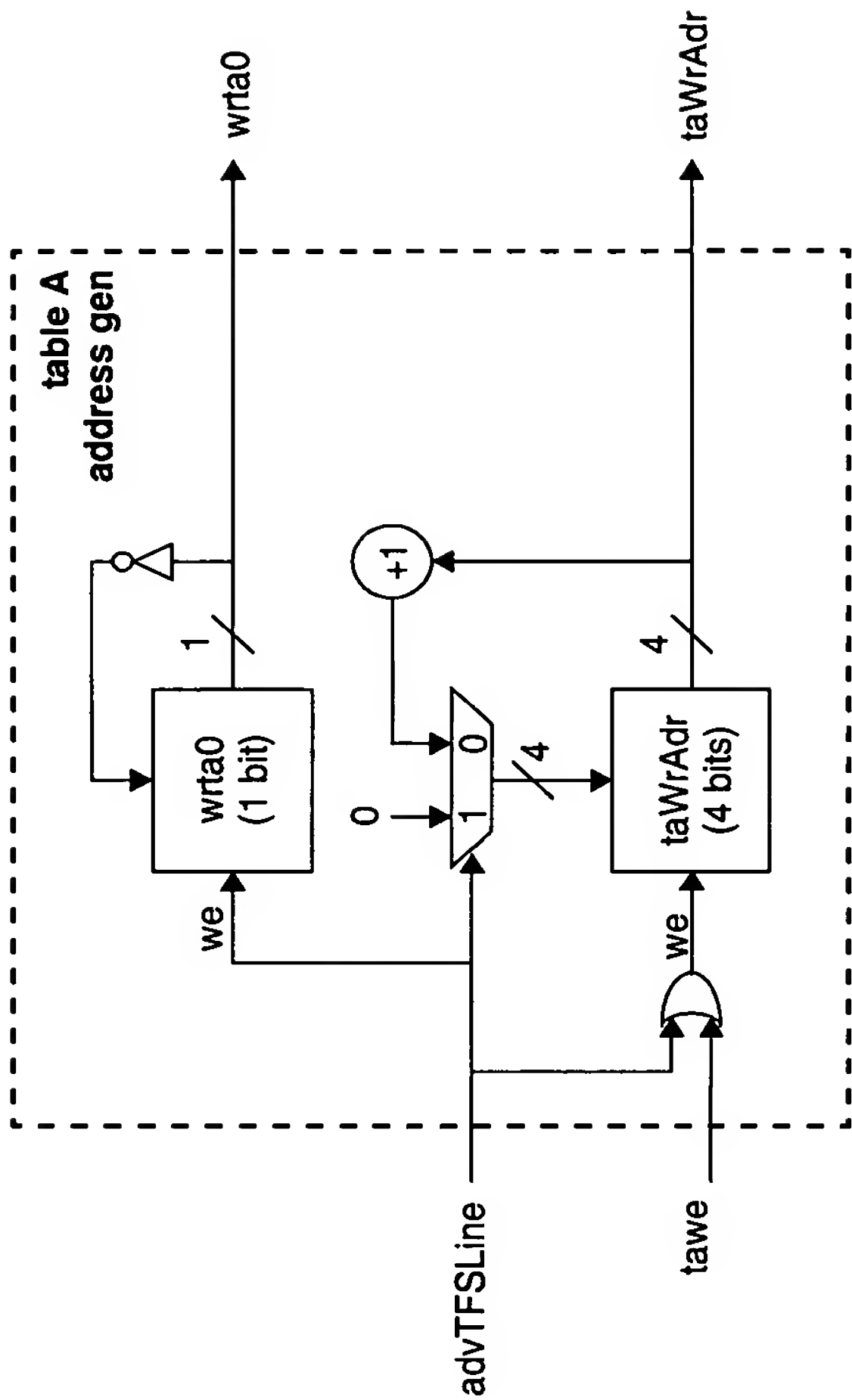


FIG. 225

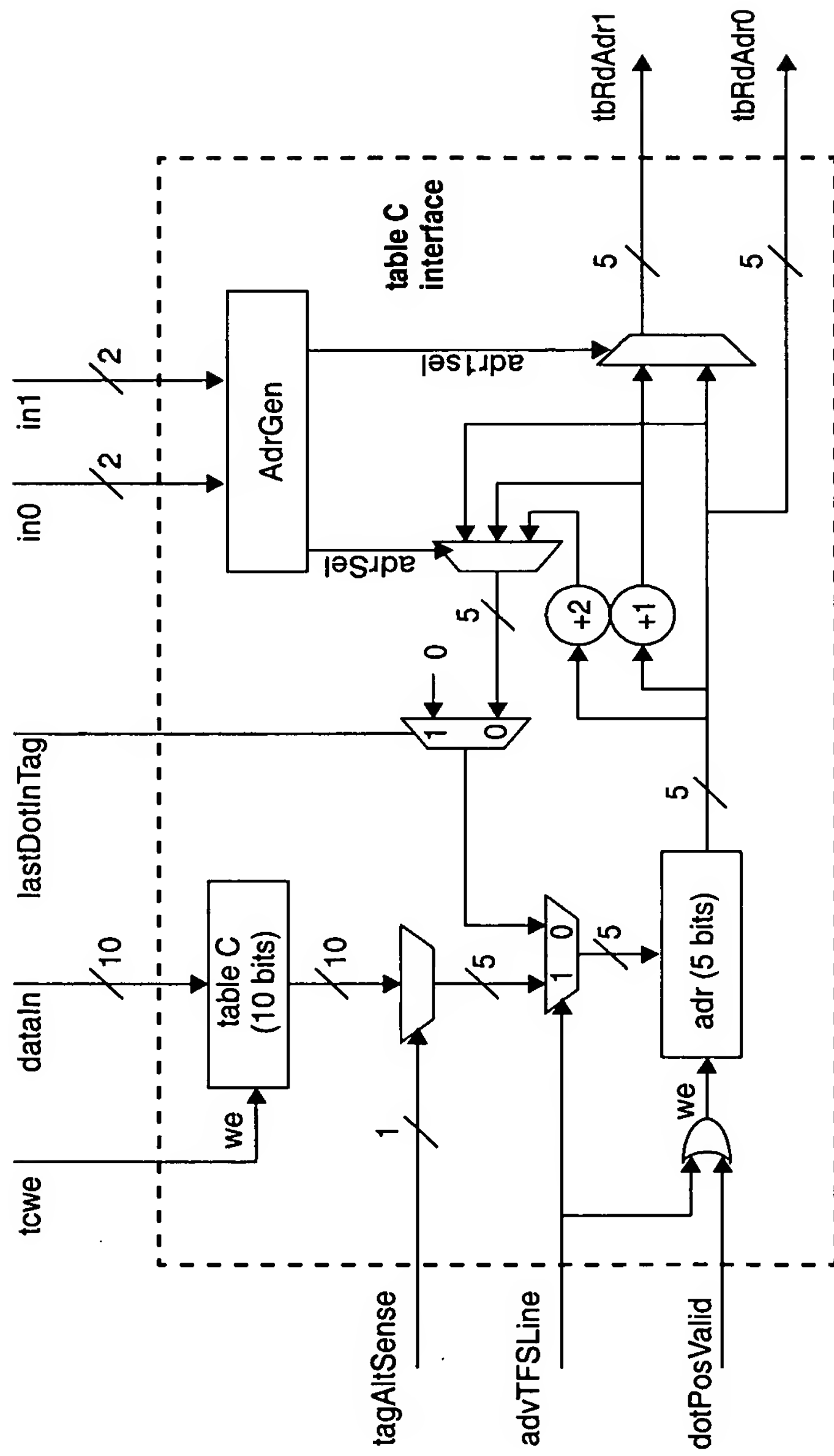


FIG. 226

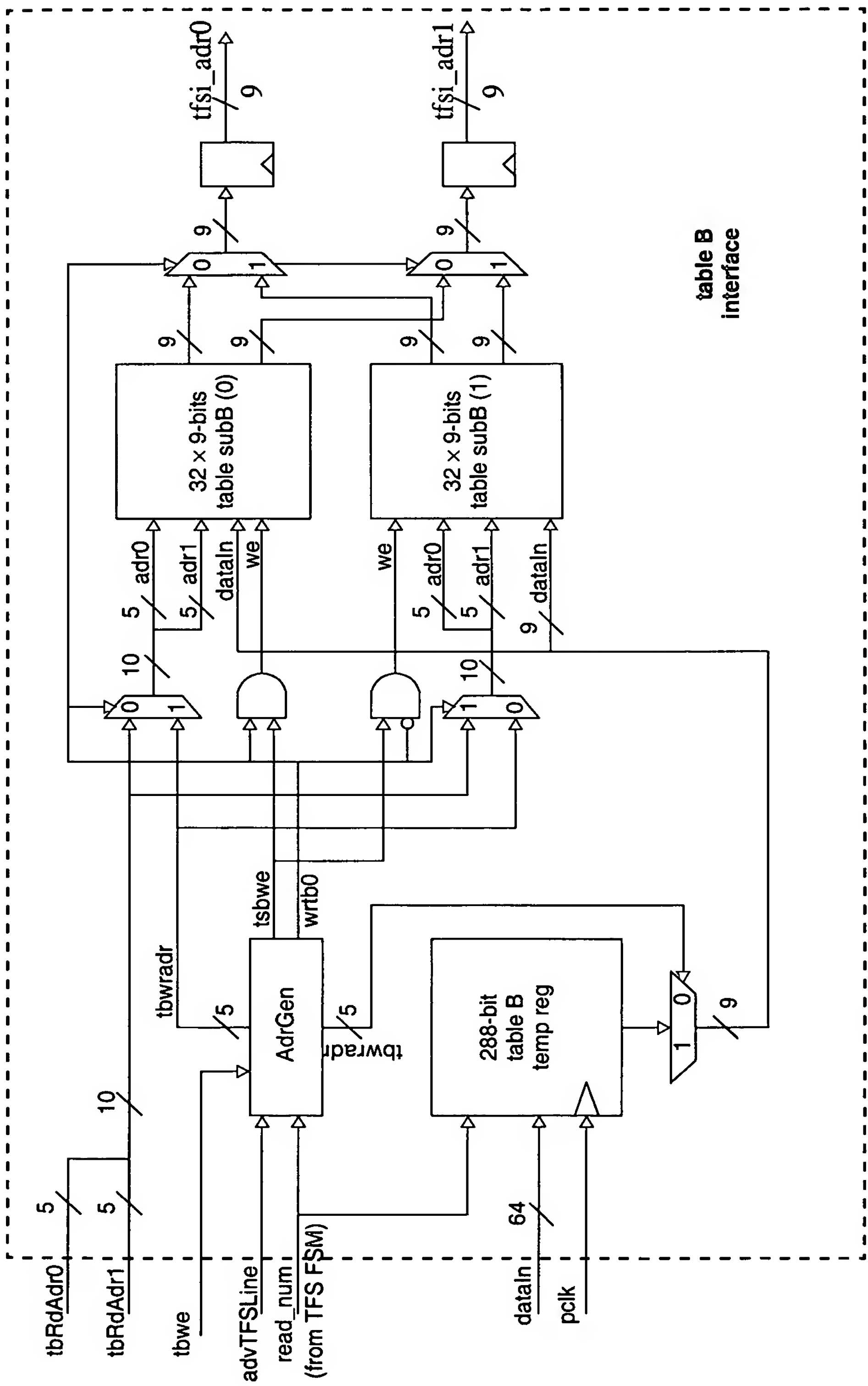
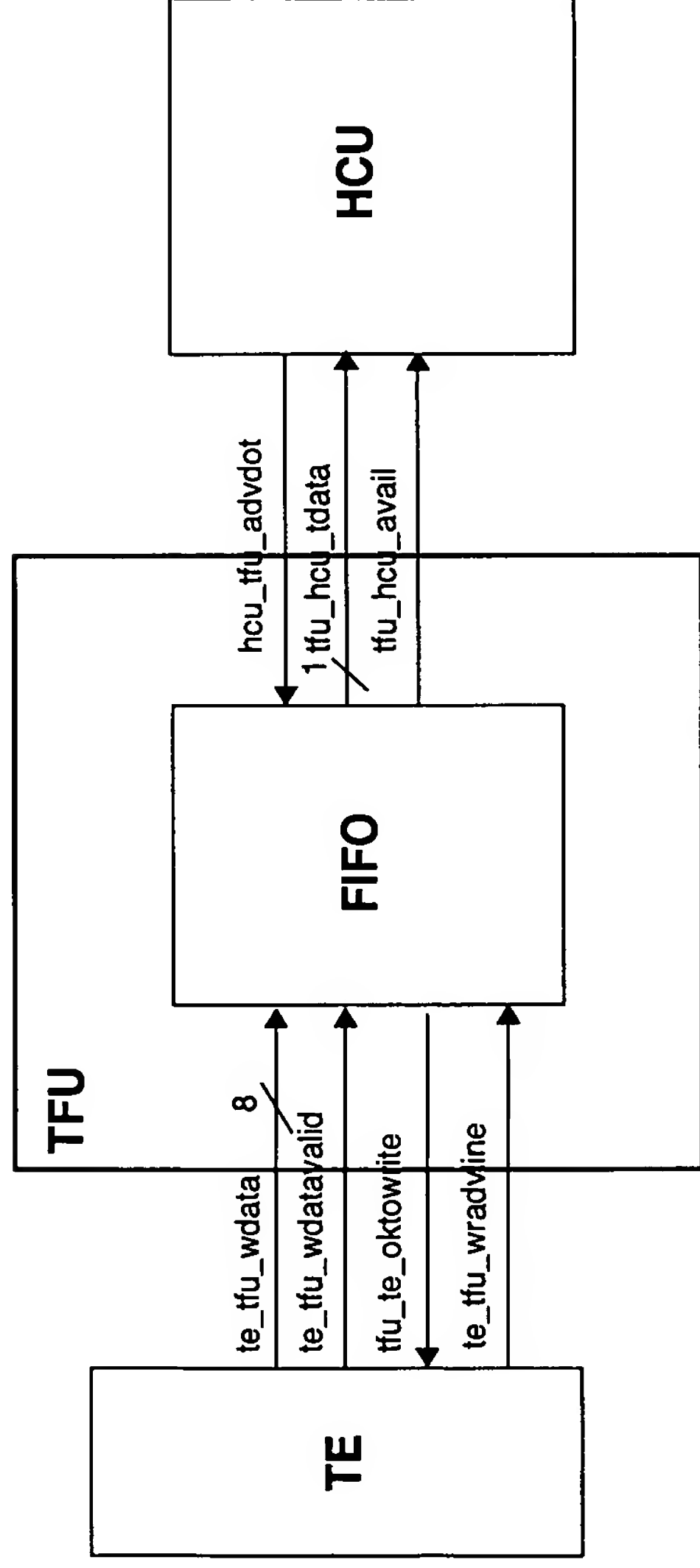


FIG. 227



195/331

FIG. 228

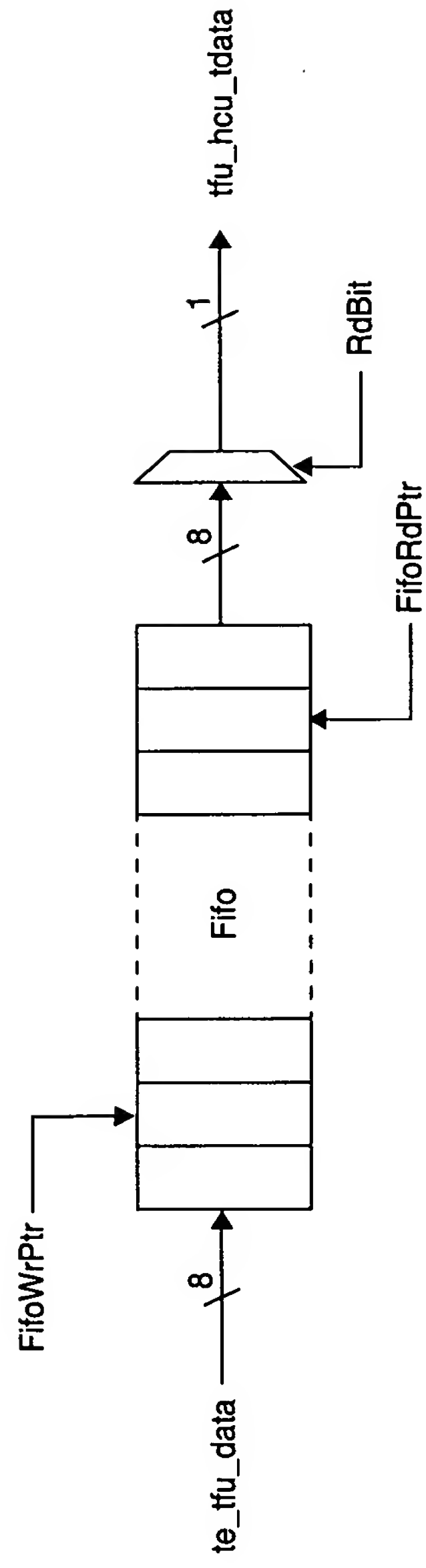


FIG. 229

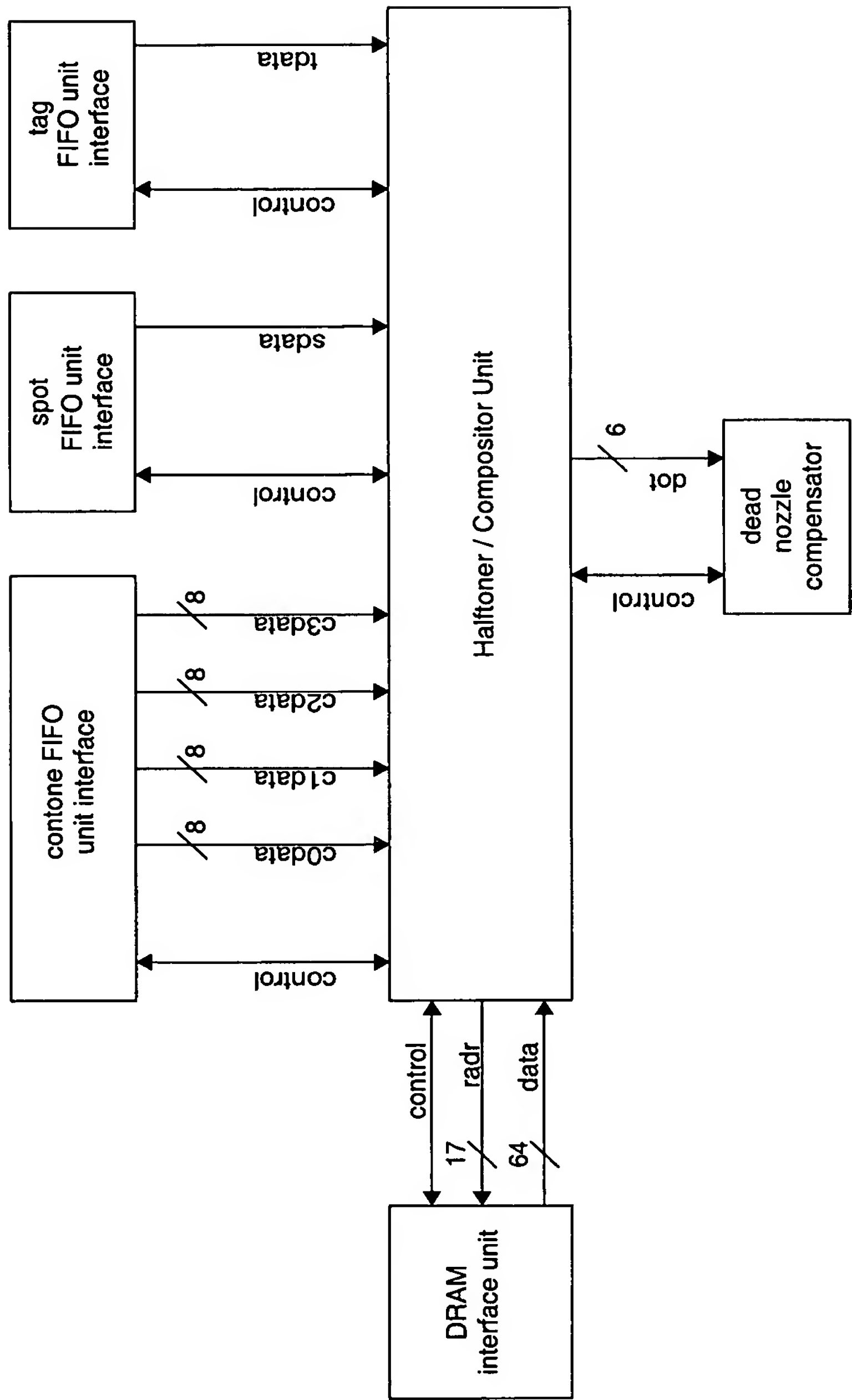


FIG. 230

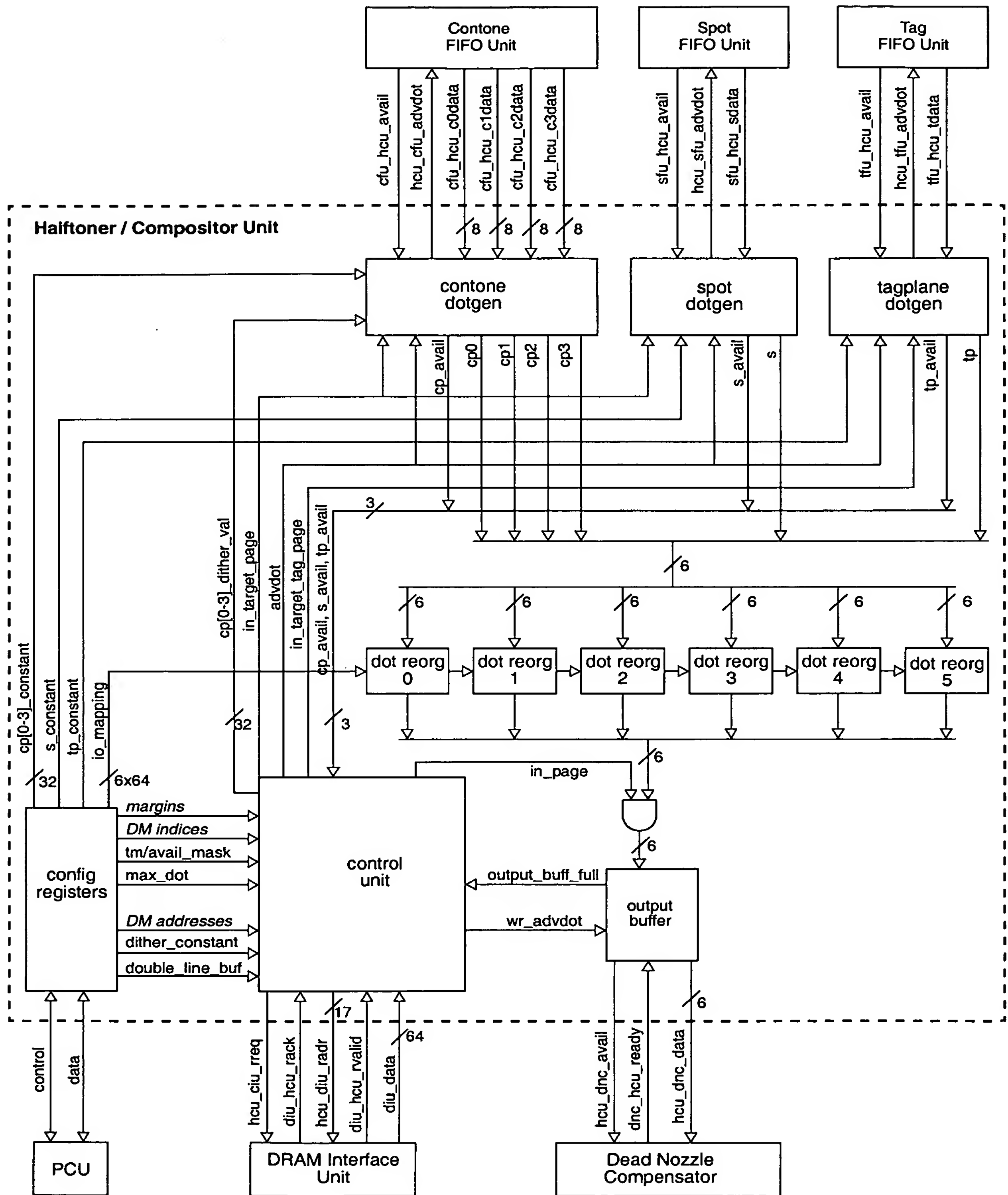
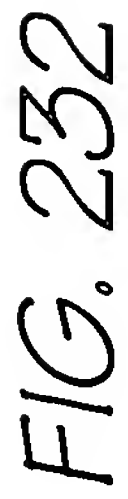


FIG. 231



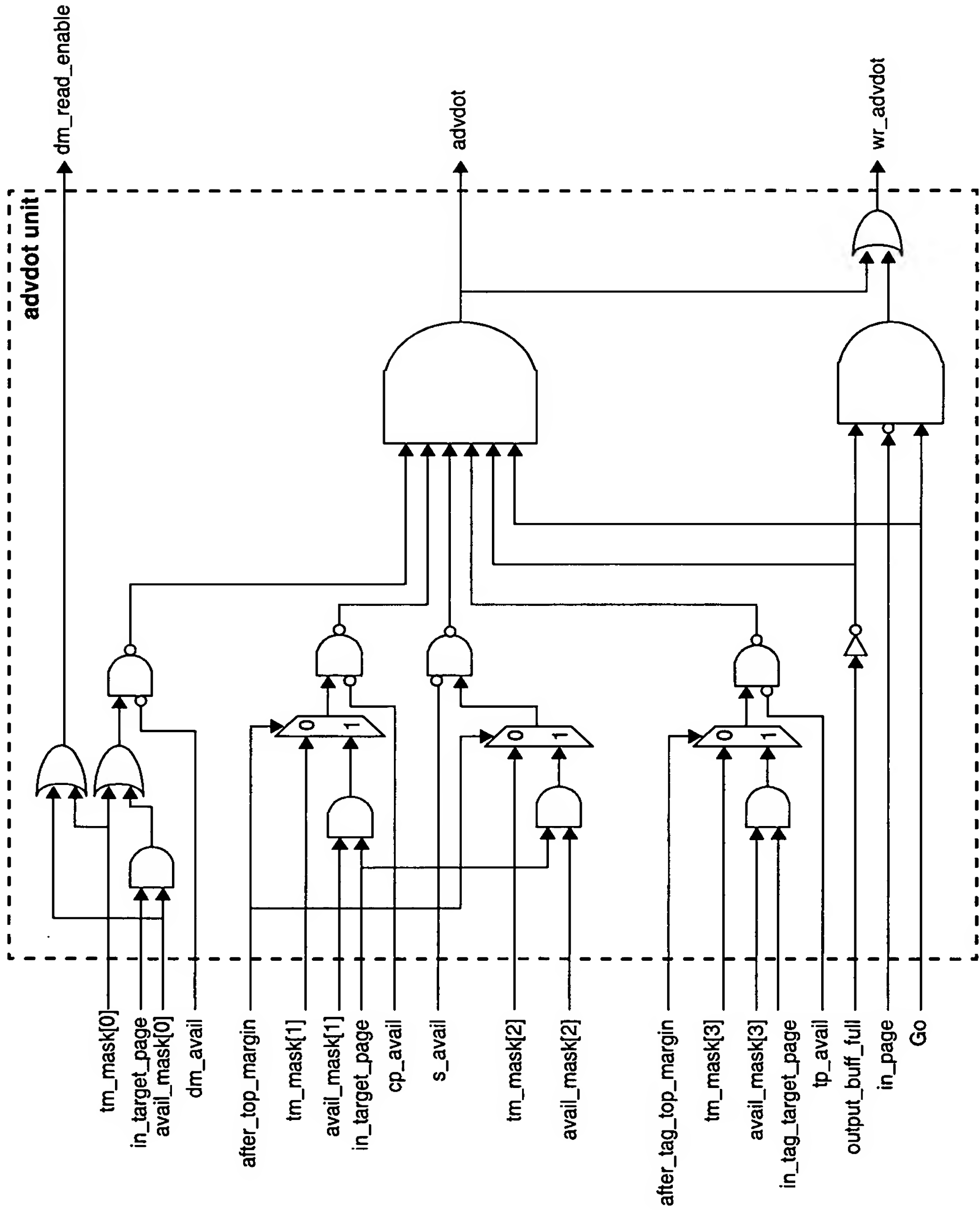


FIG. 233

200/331

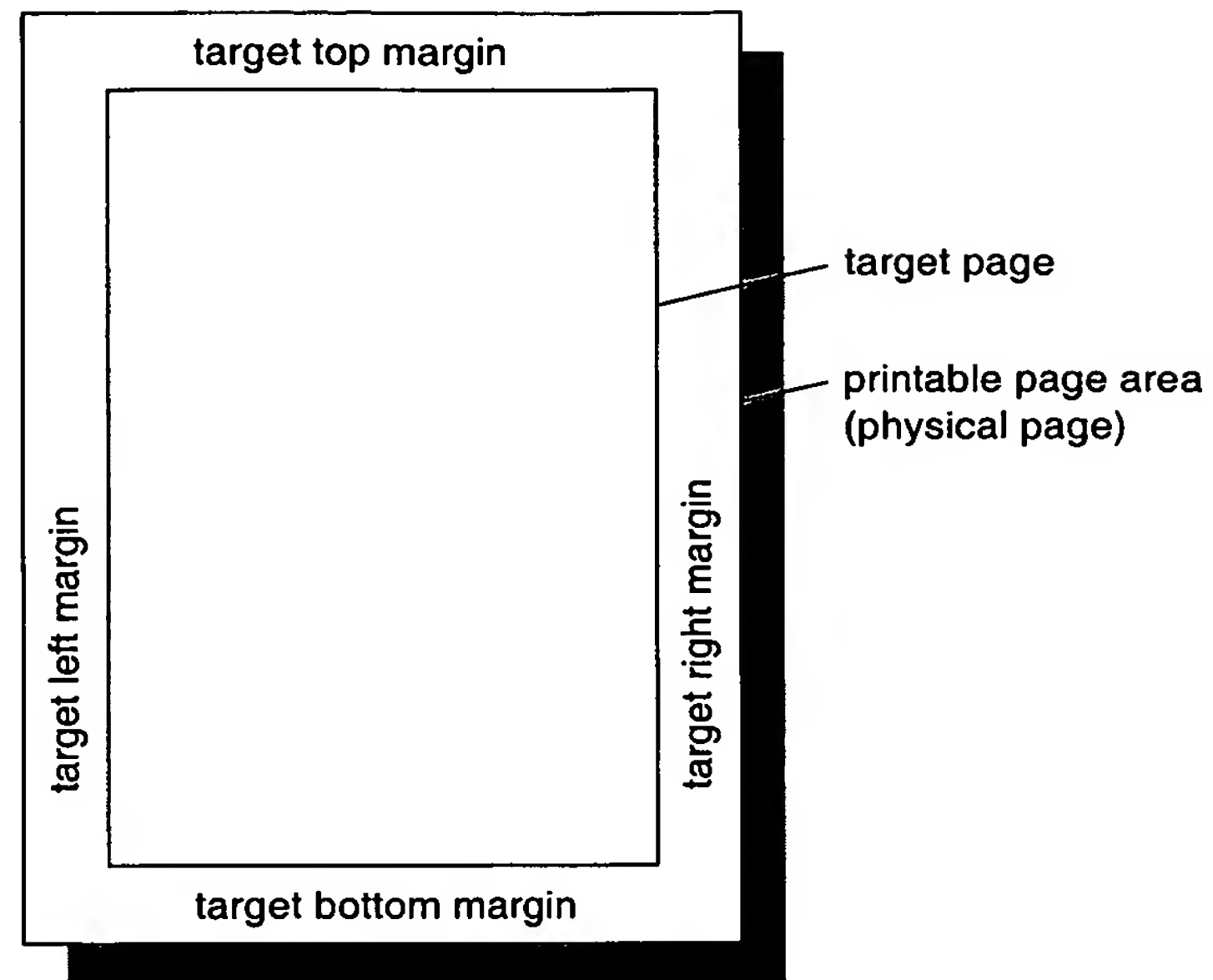


FIG. 234

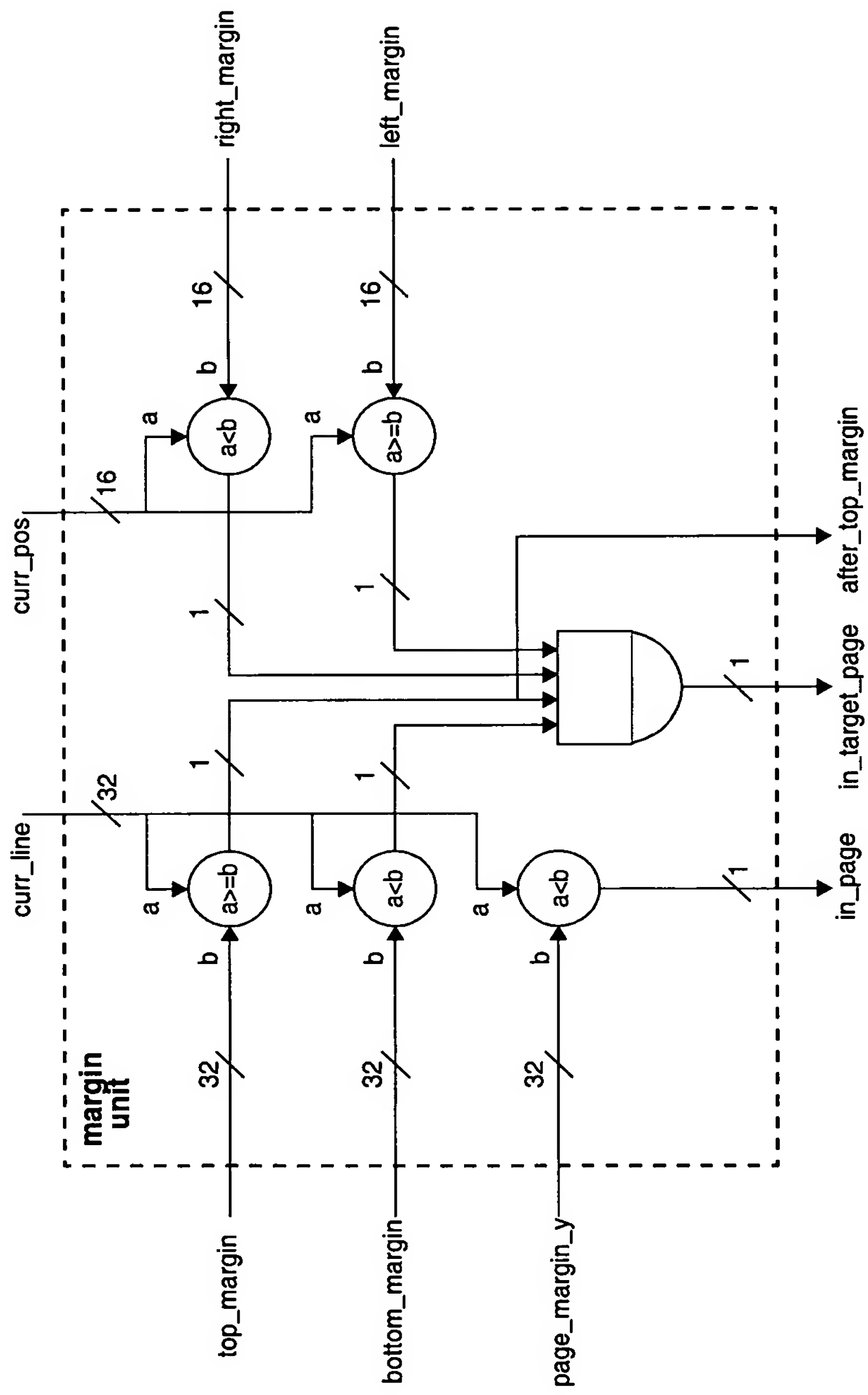


FIG. 235

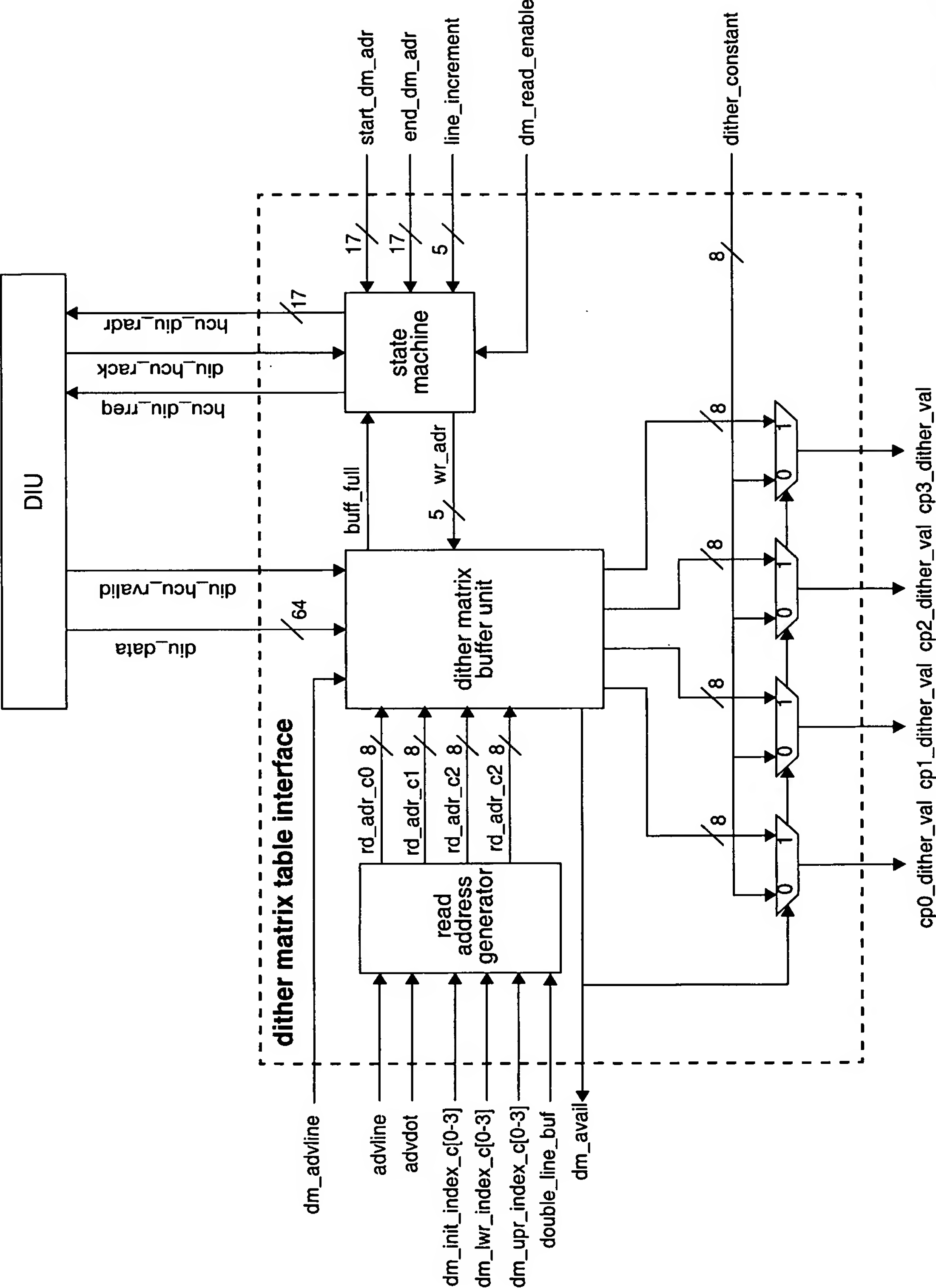


FIG. 236

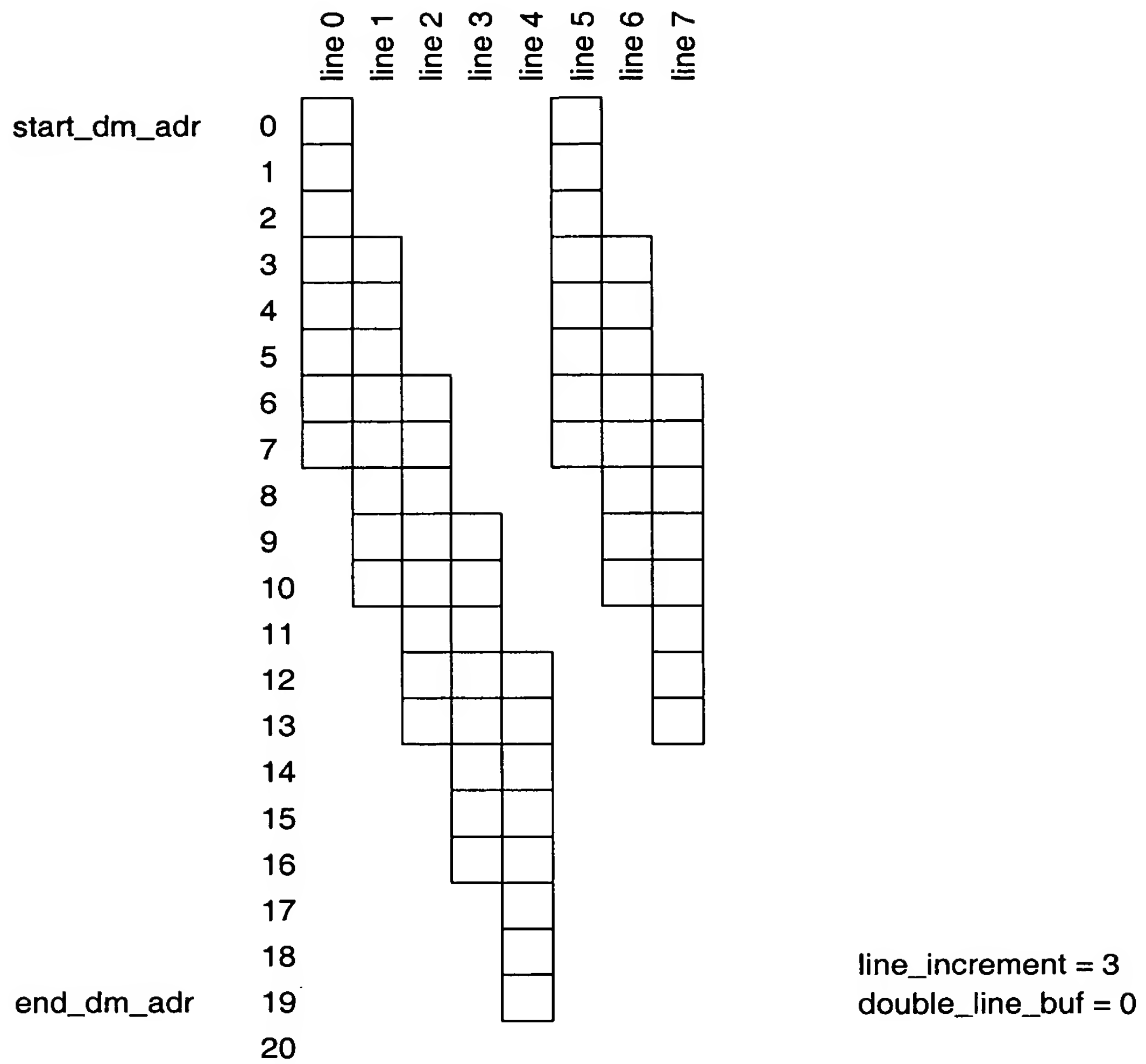


FIG. 237

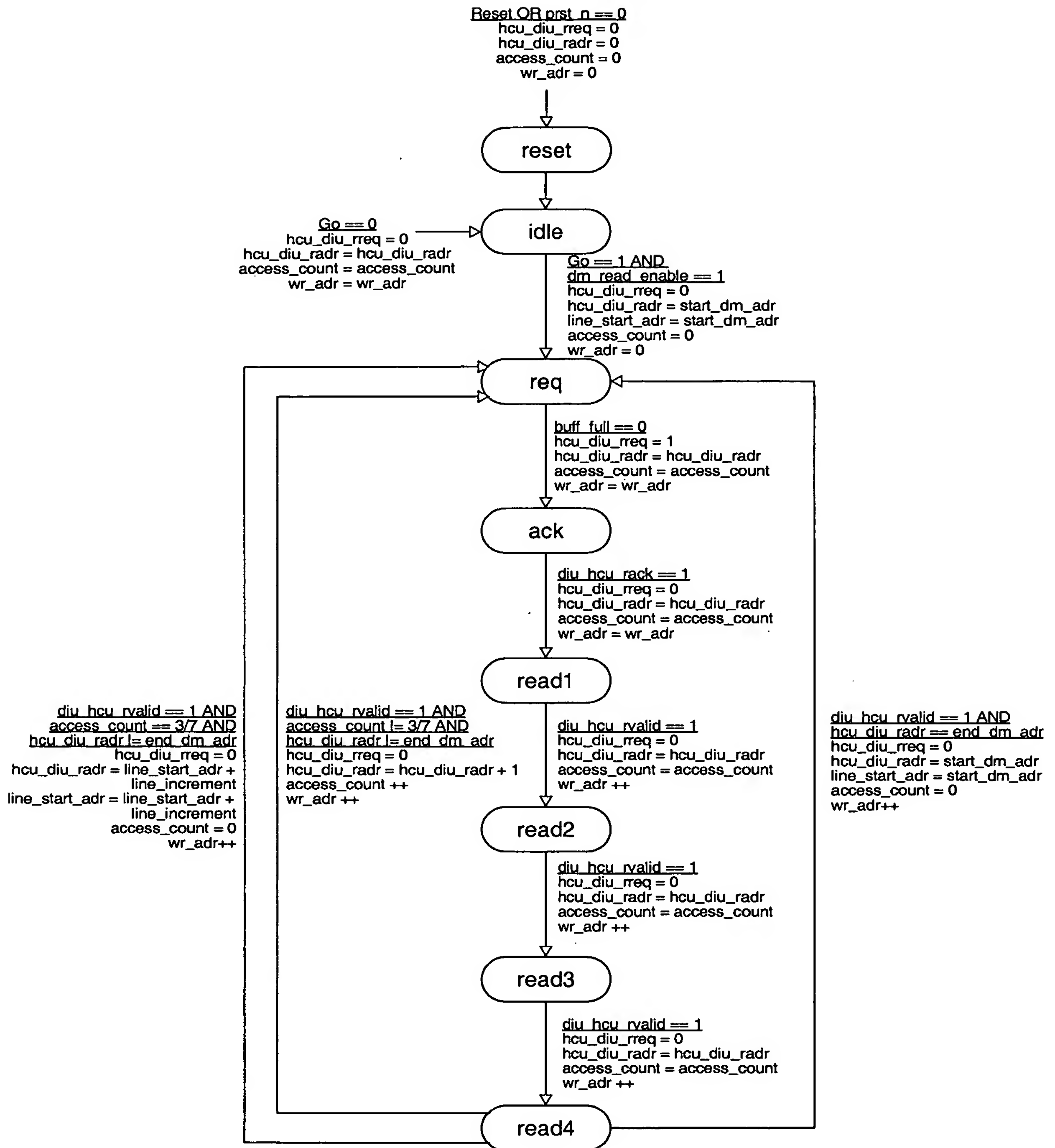


FIG. 238

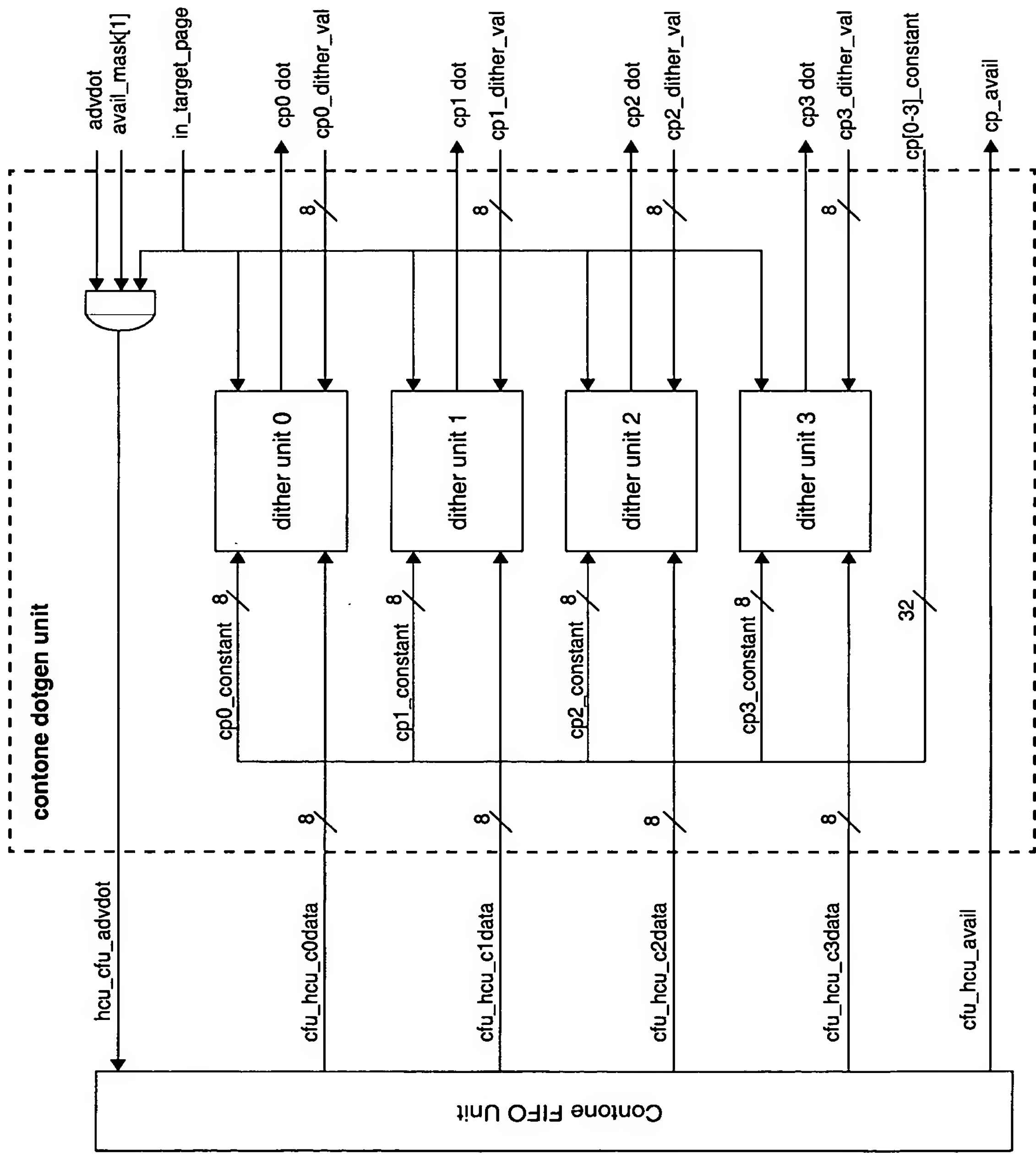


FIG. 239

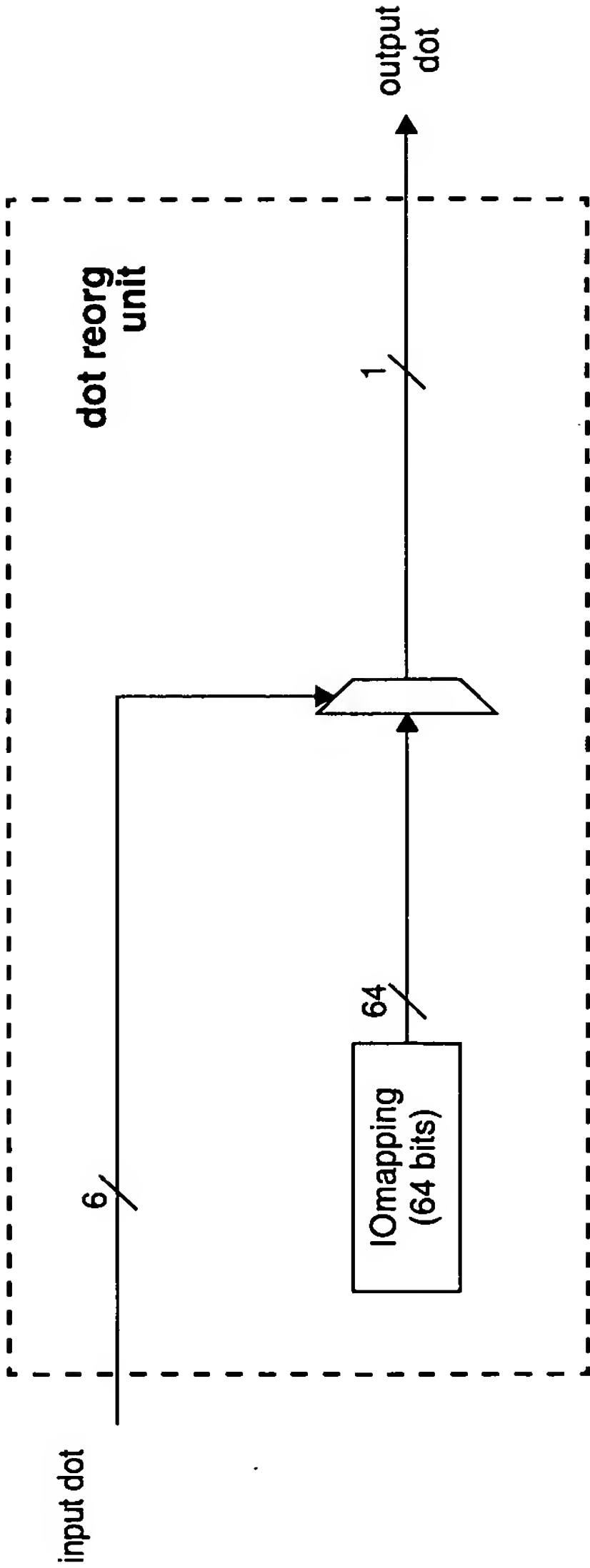
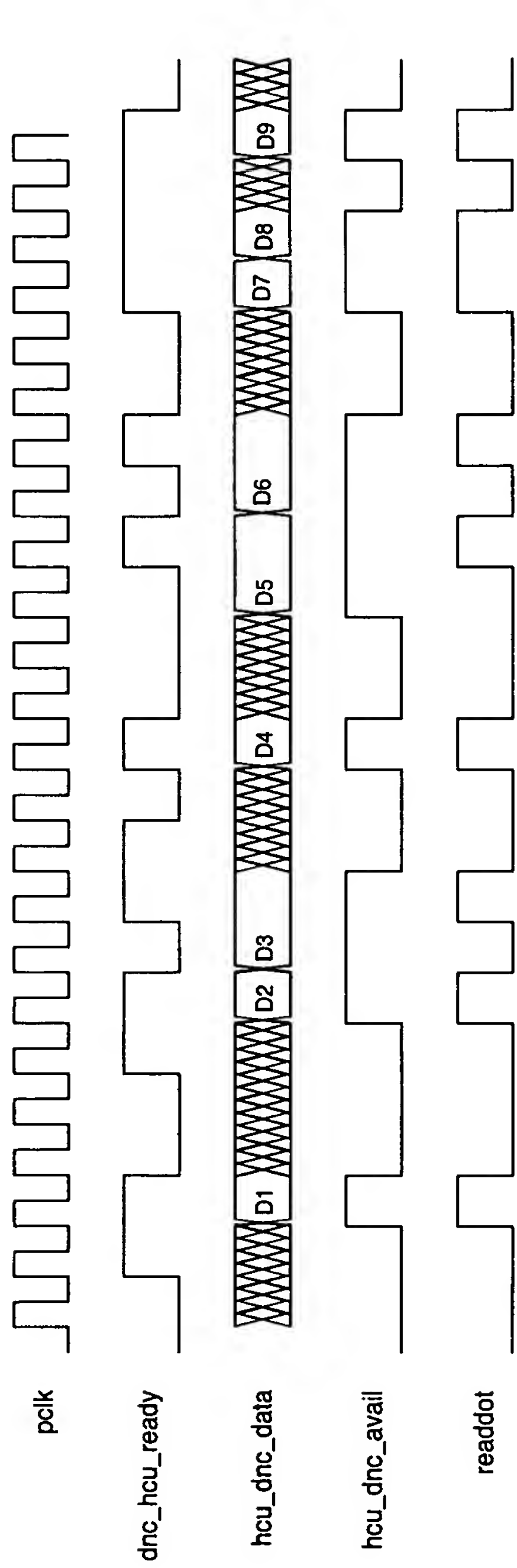


FIG. 240



207/331

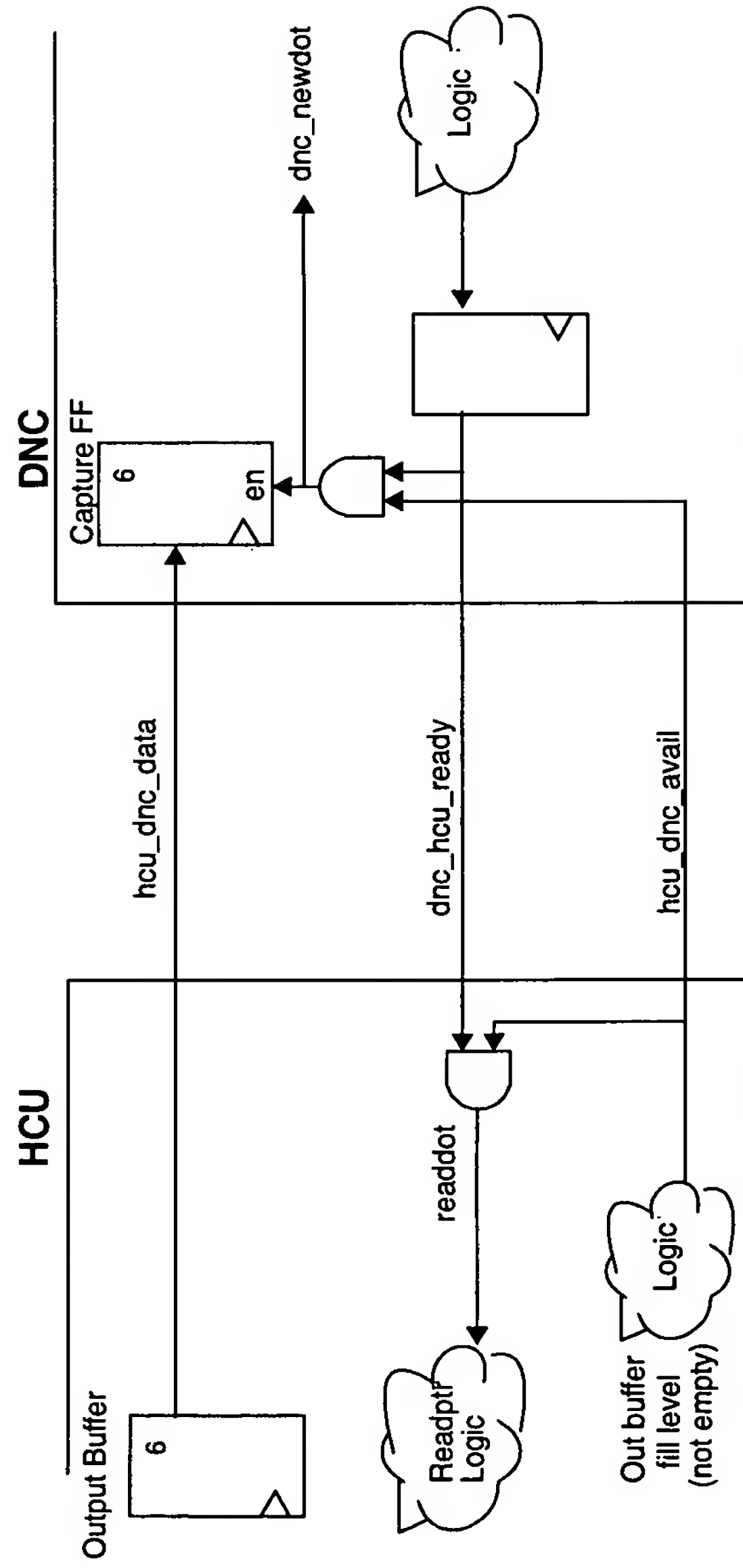


FIG. 241

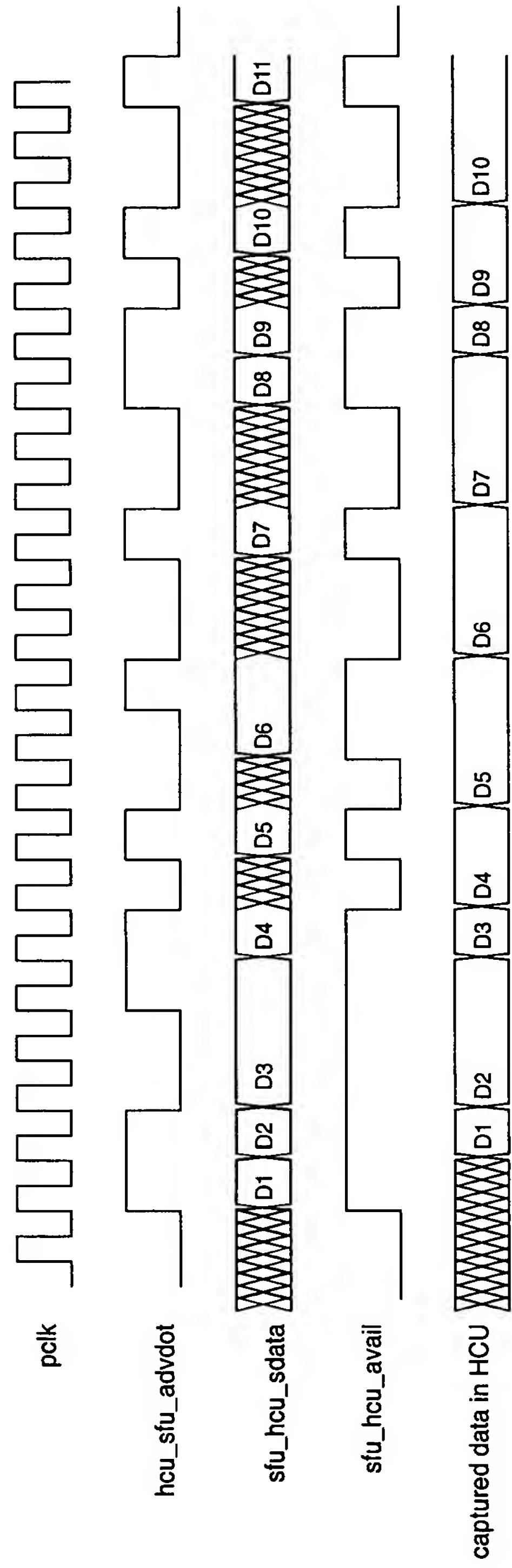


FIG. 242

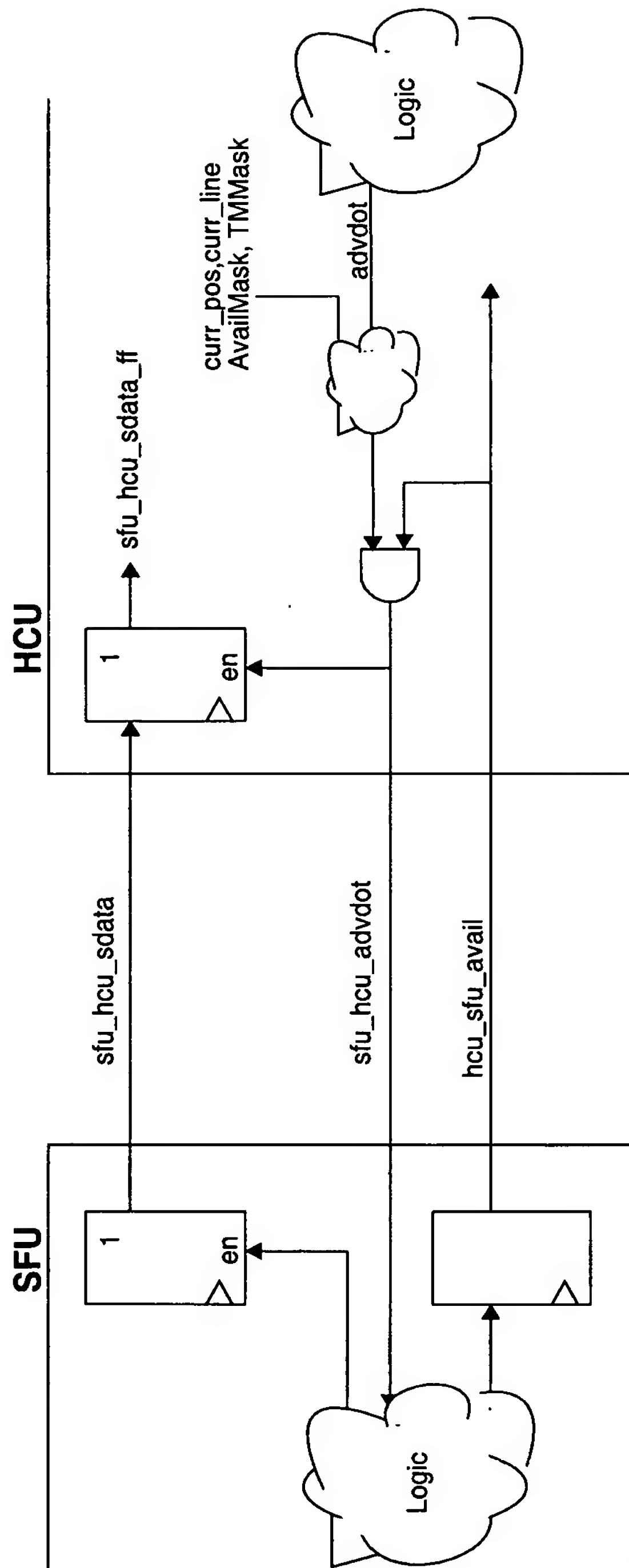


FIG. 243

210/331

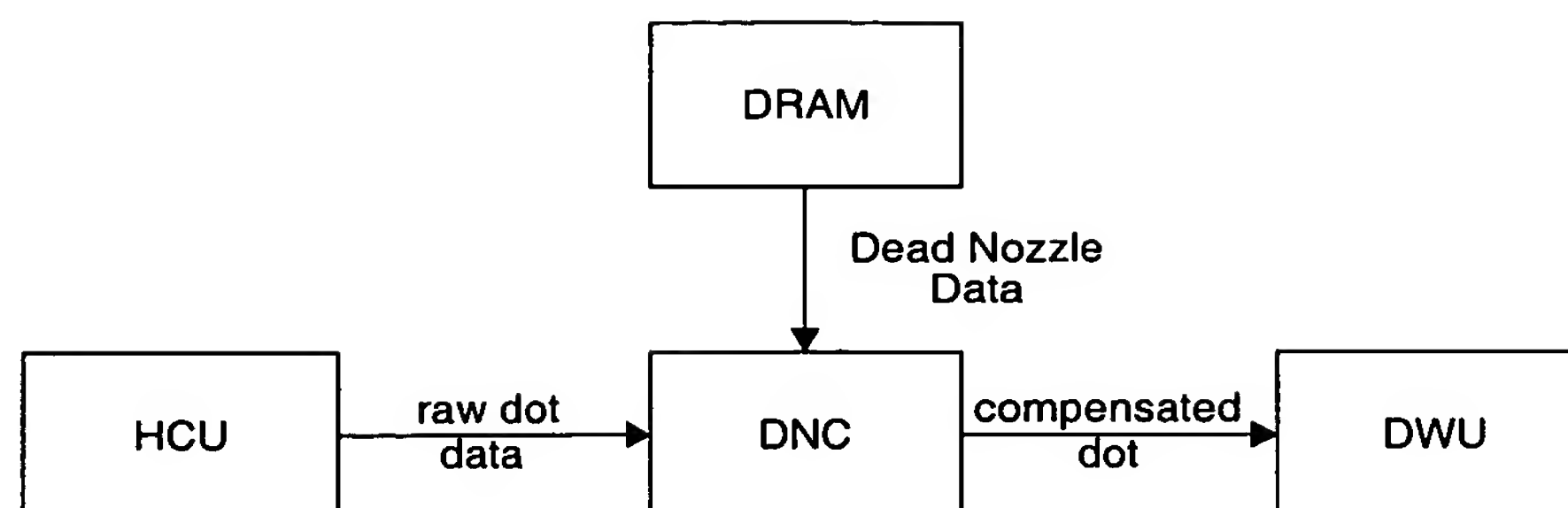


FIG. 244

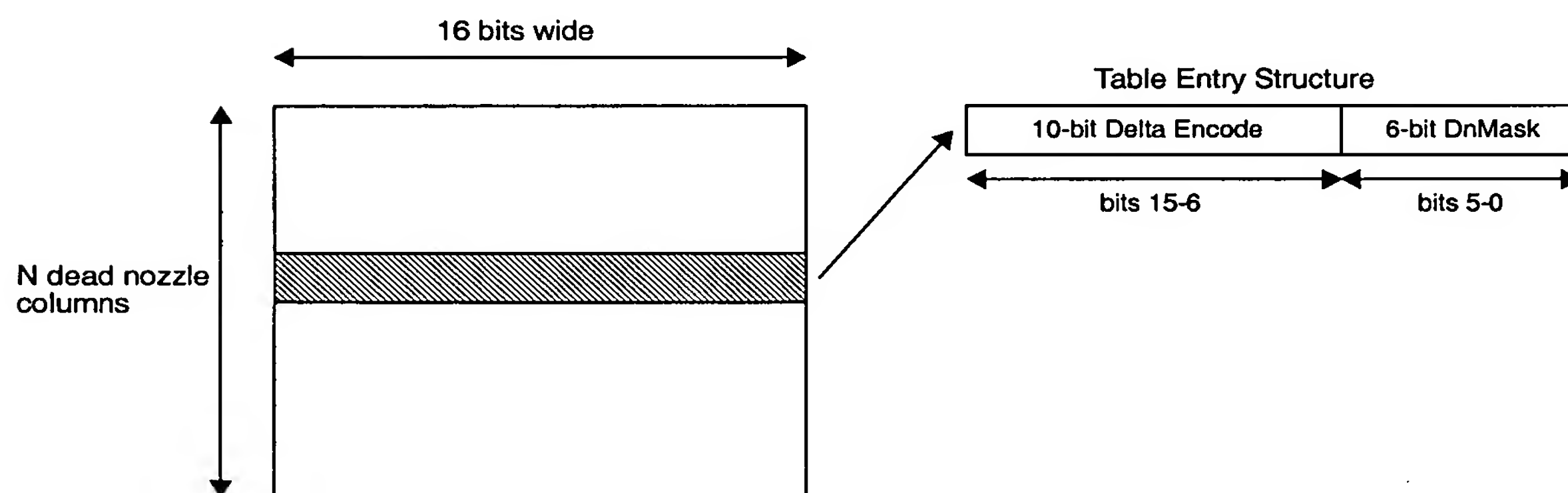


FIG. 245

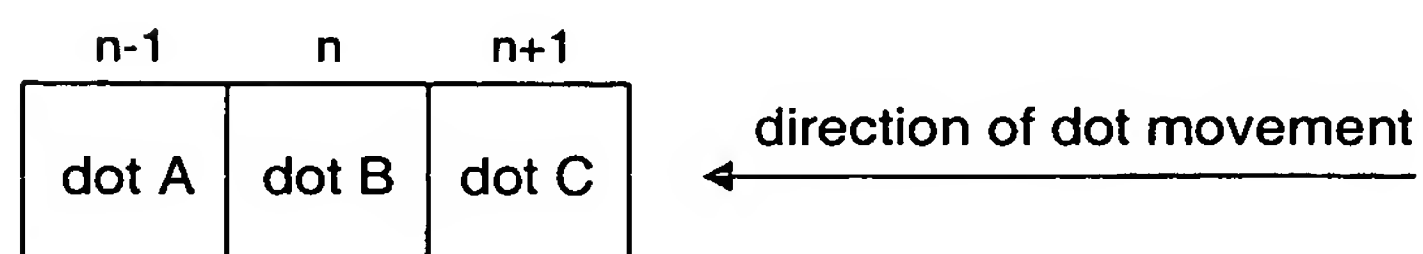


FIG. 246

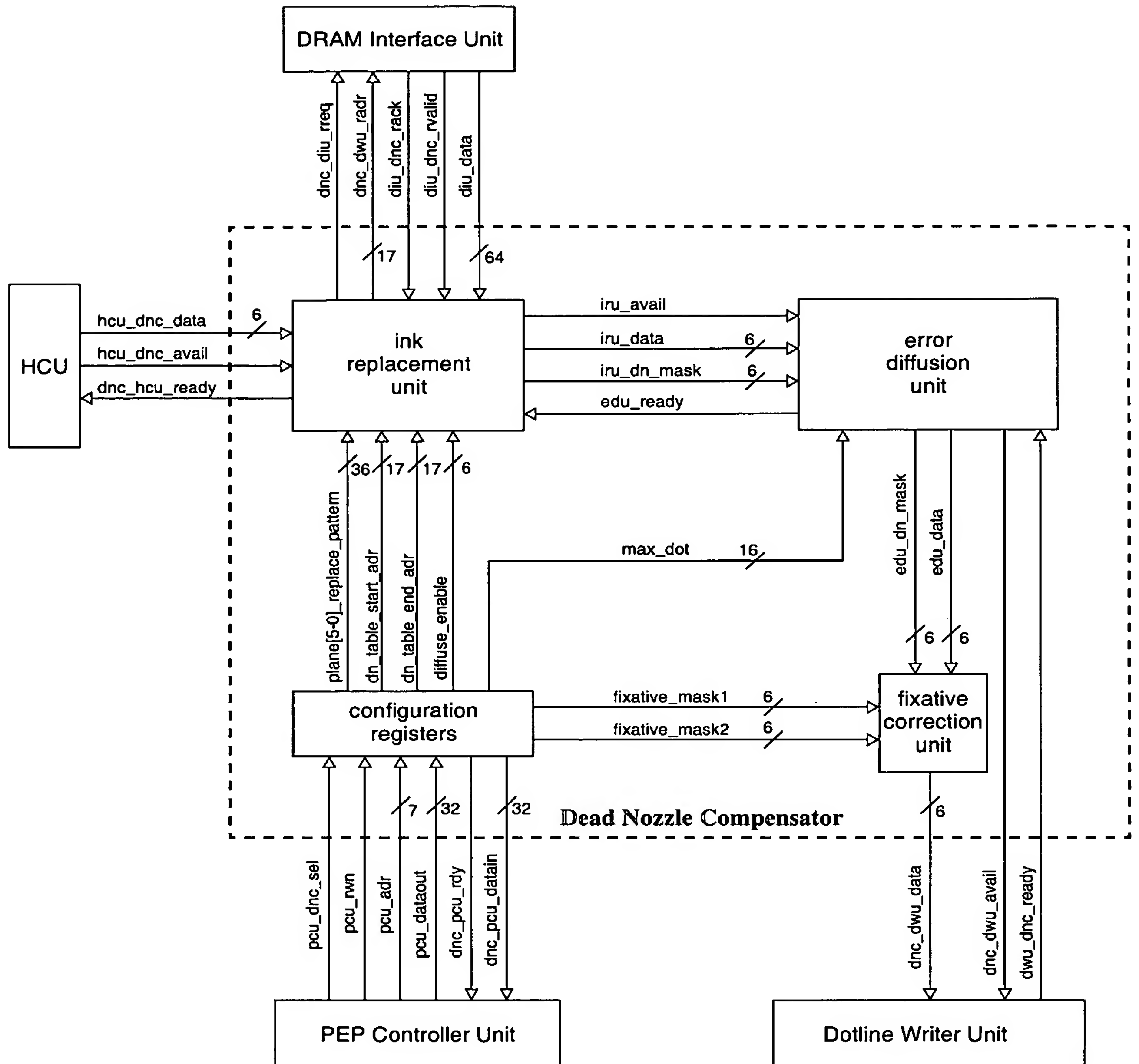


FIG. 247

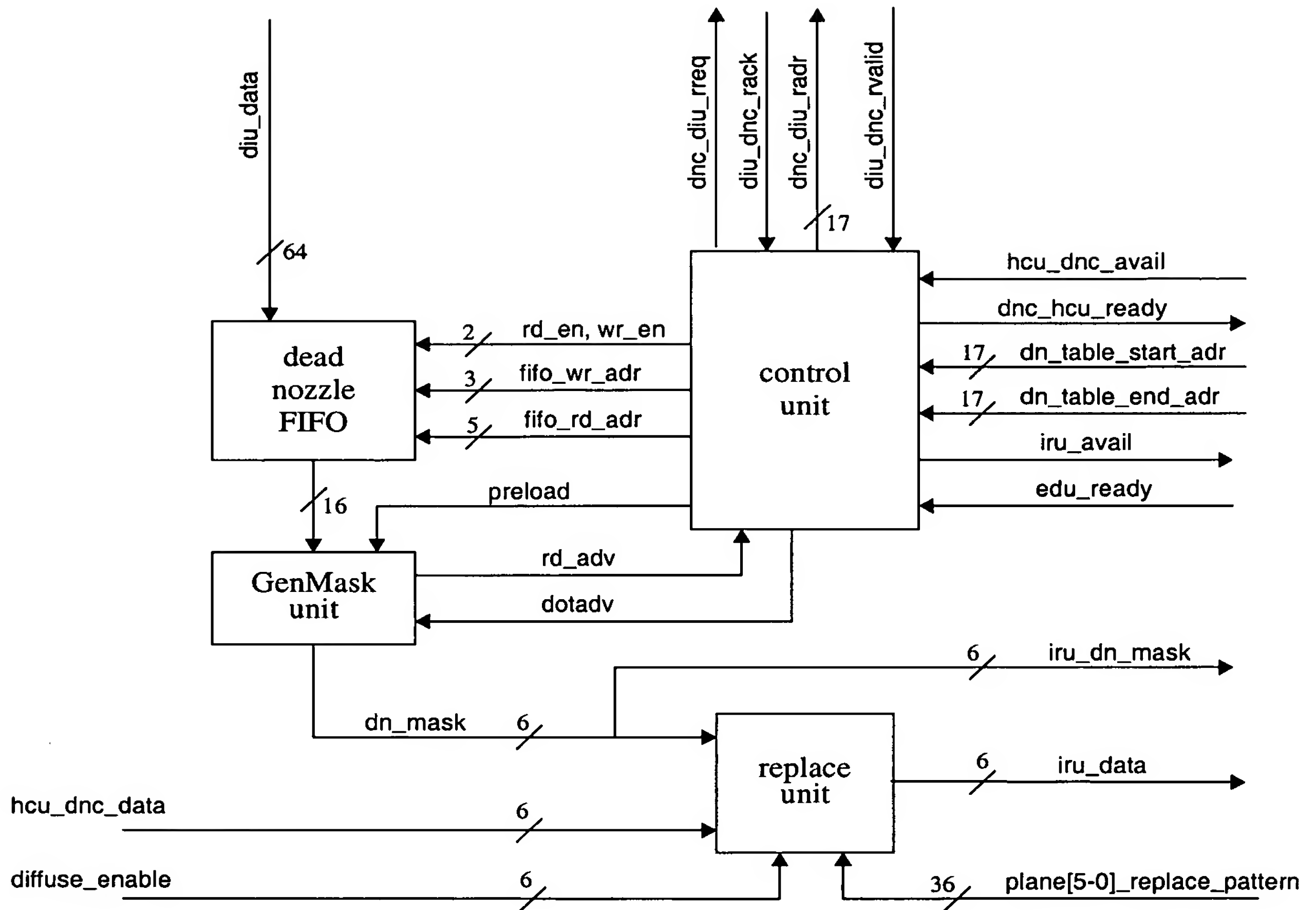


FIG. 248

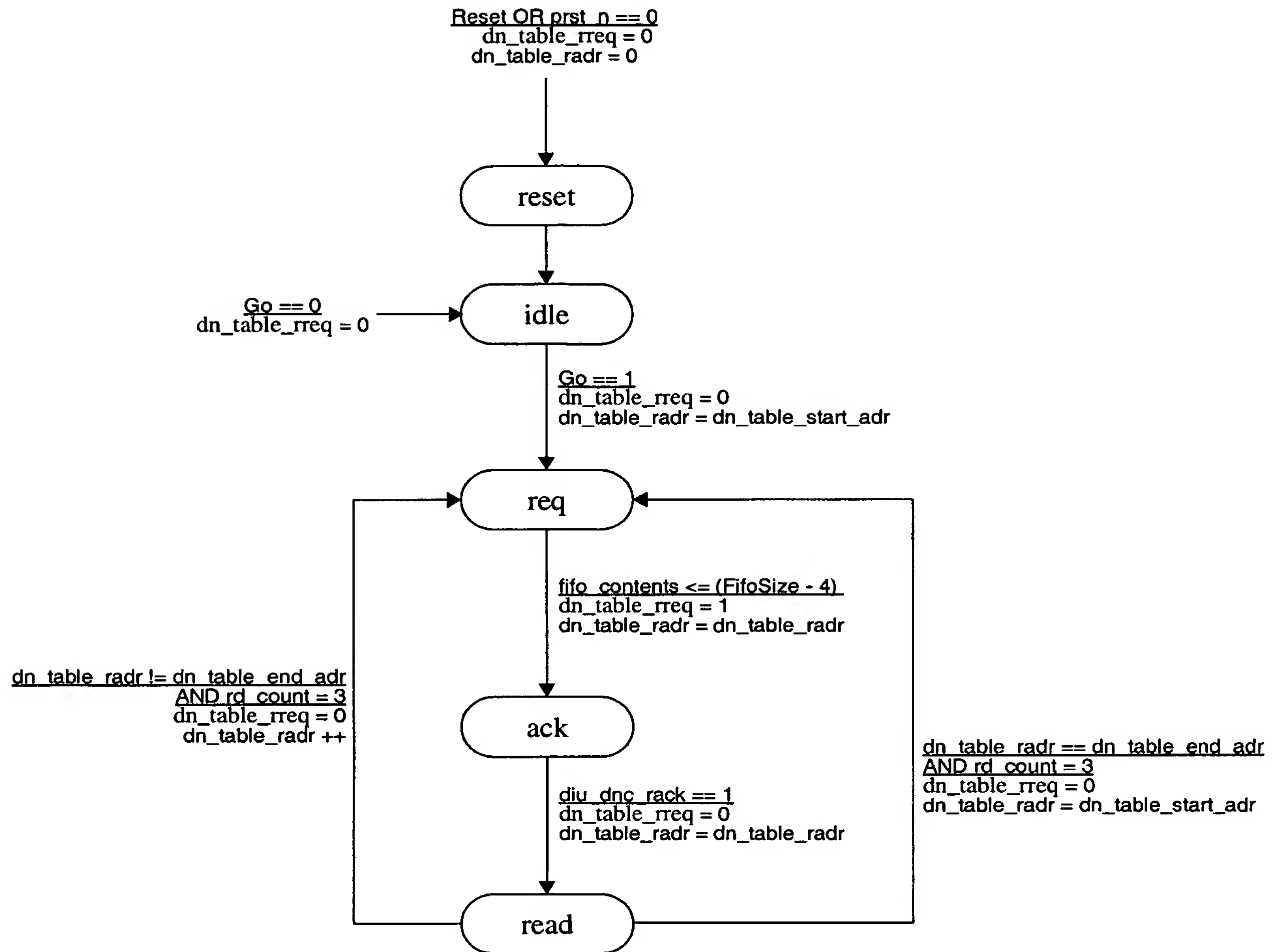


FIG. 249

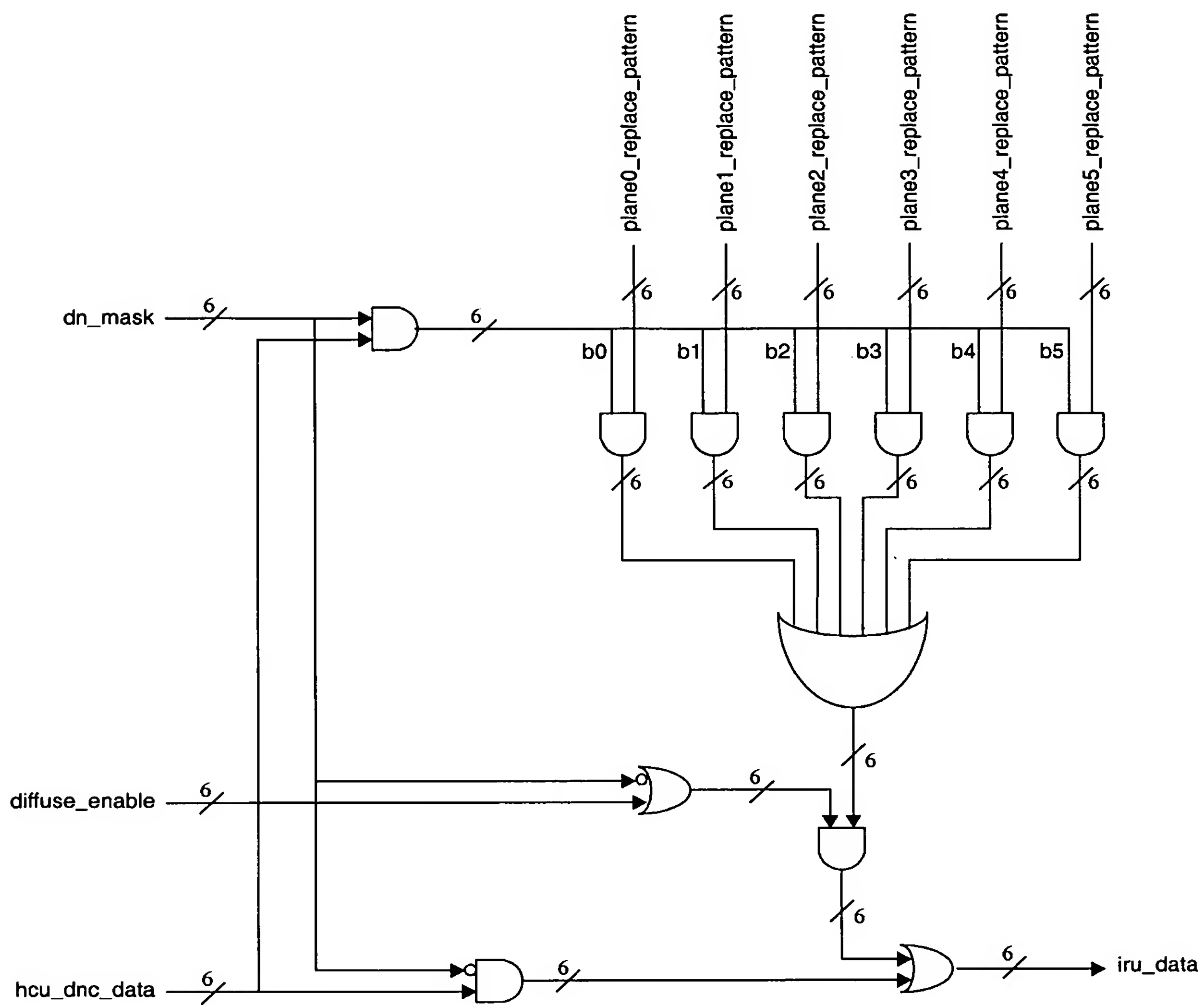


FIG. 250

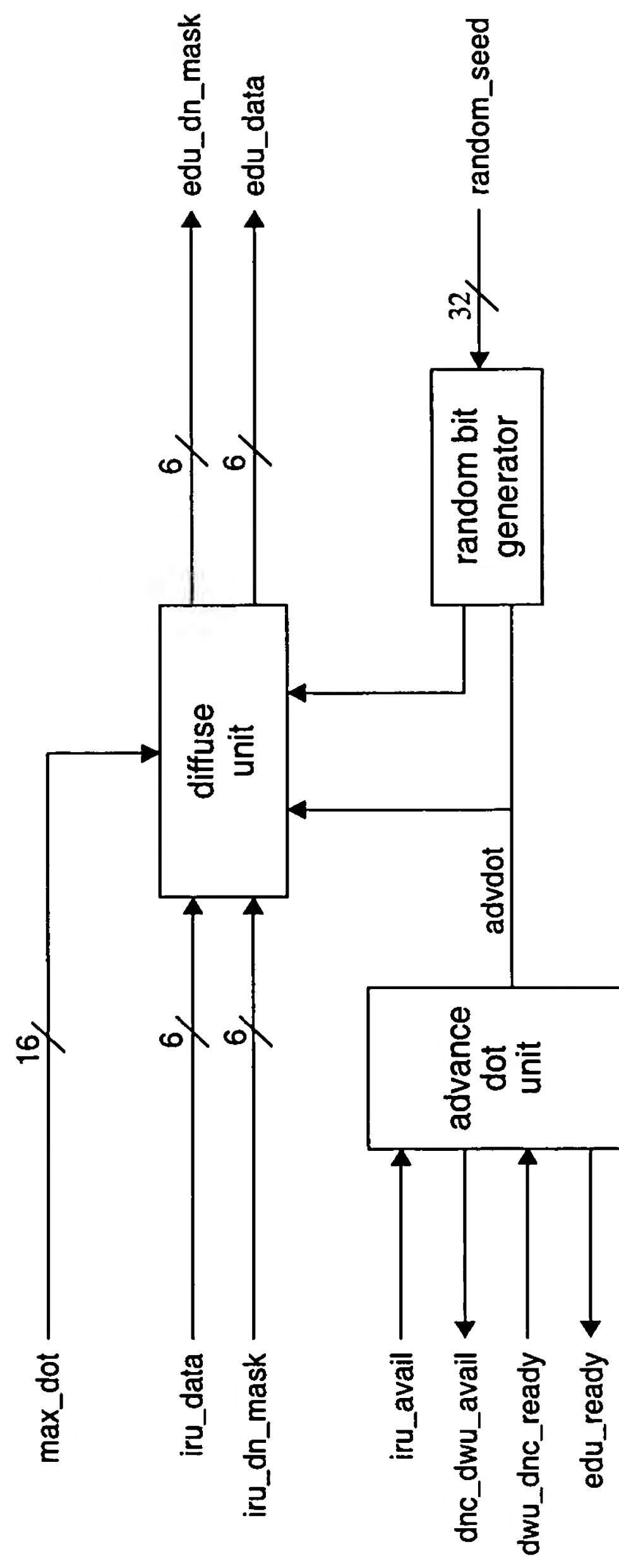


FIG. 251

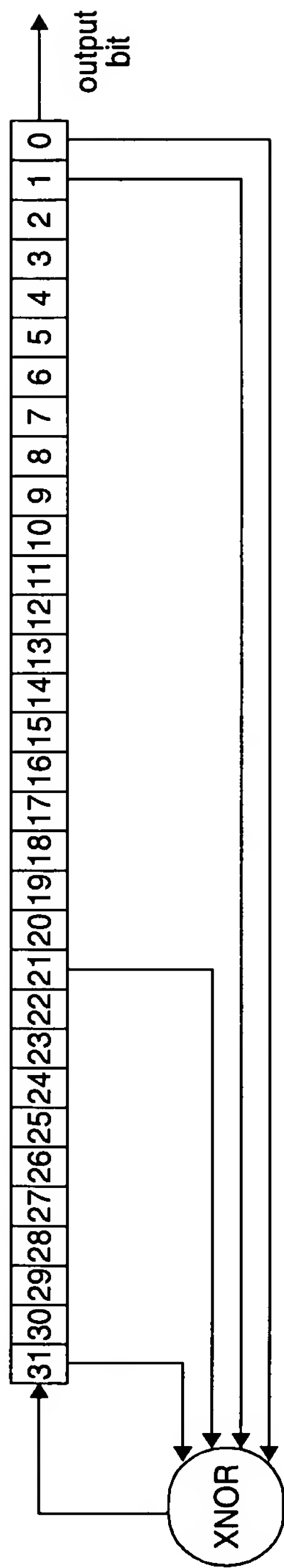


FIG. 252

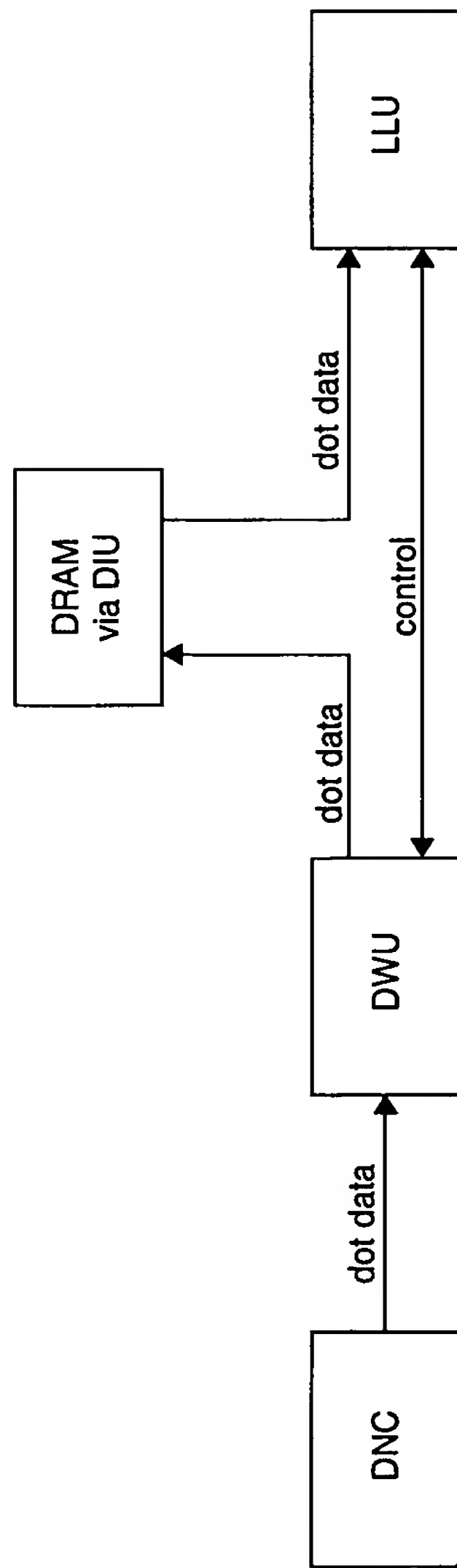
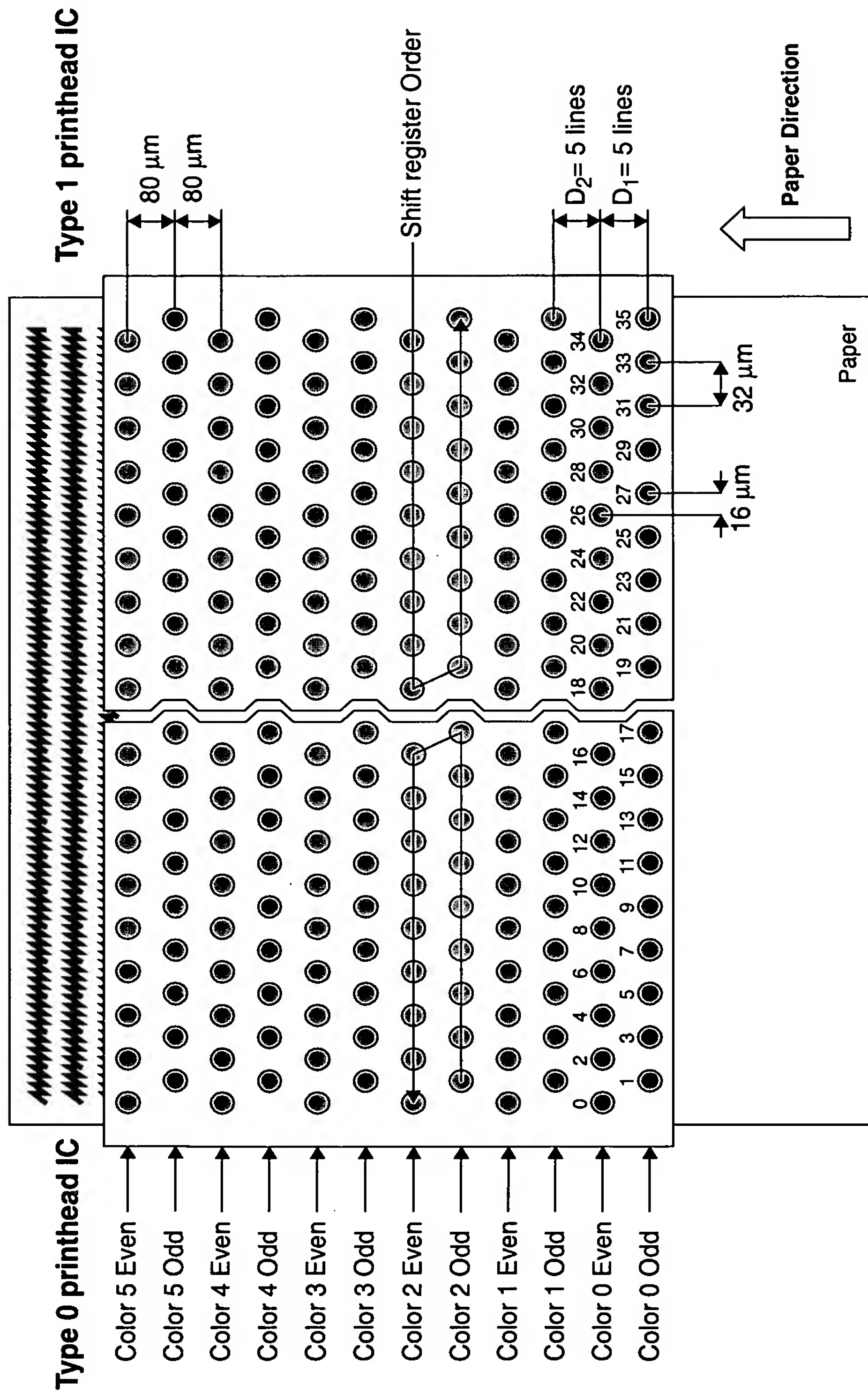


FIG. 253



Note: Paper passes under printhead

FIG. 254

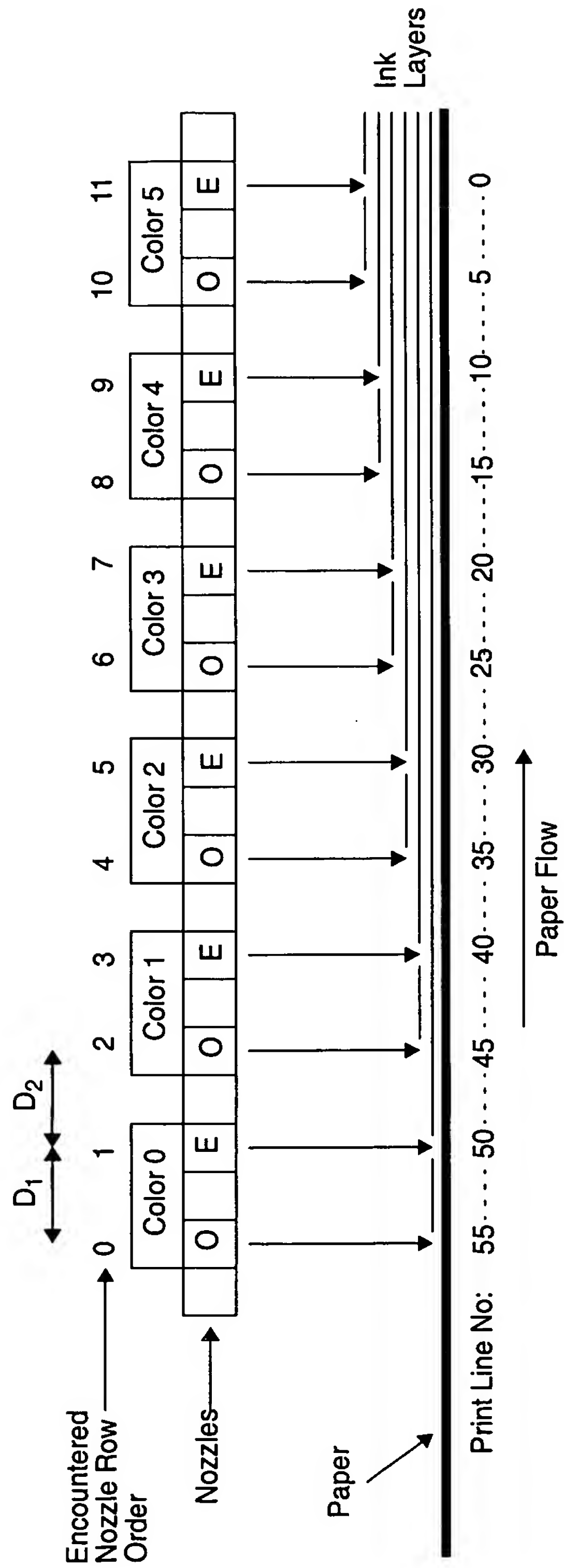


FIG. 255

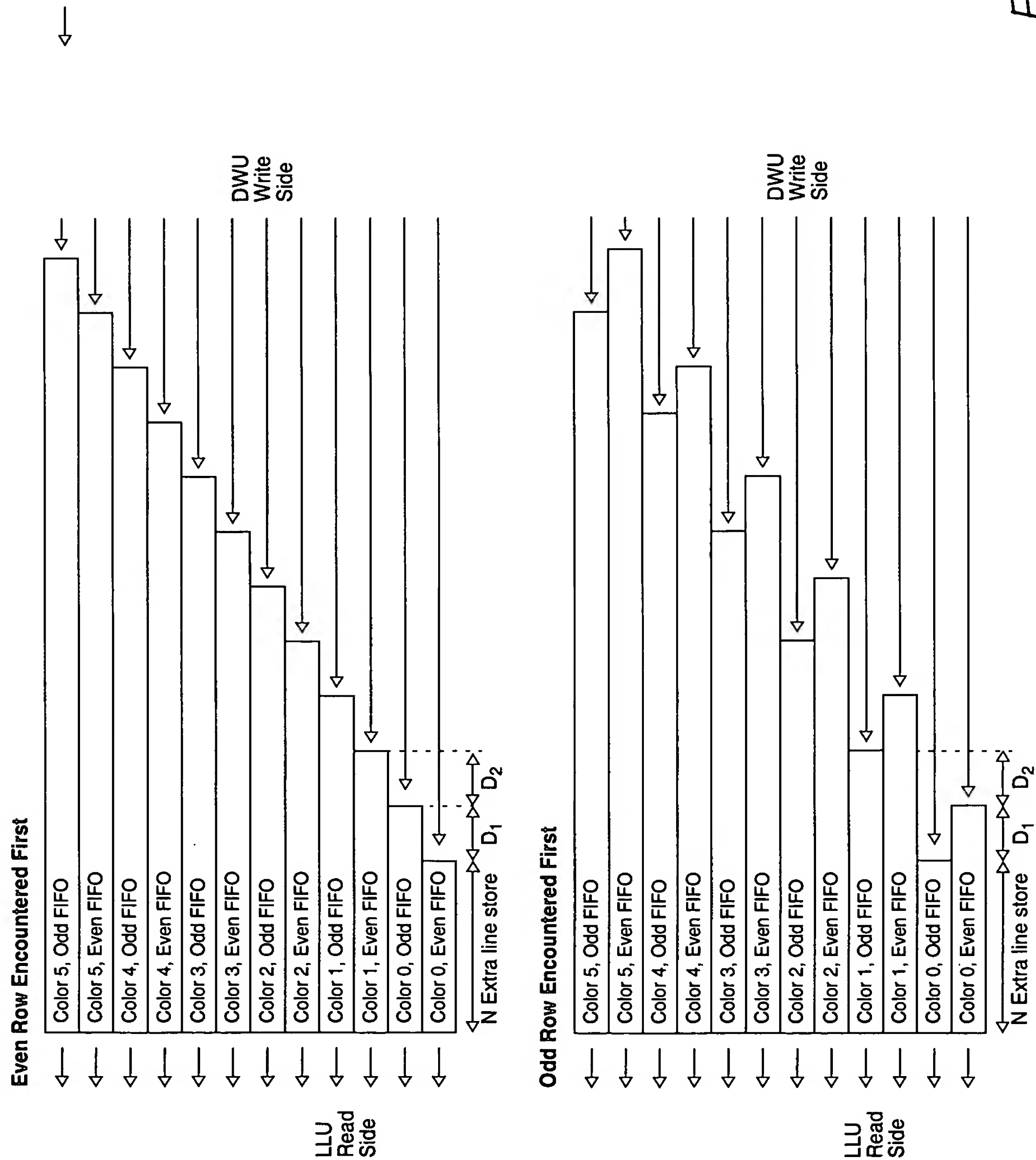


FIG. 256

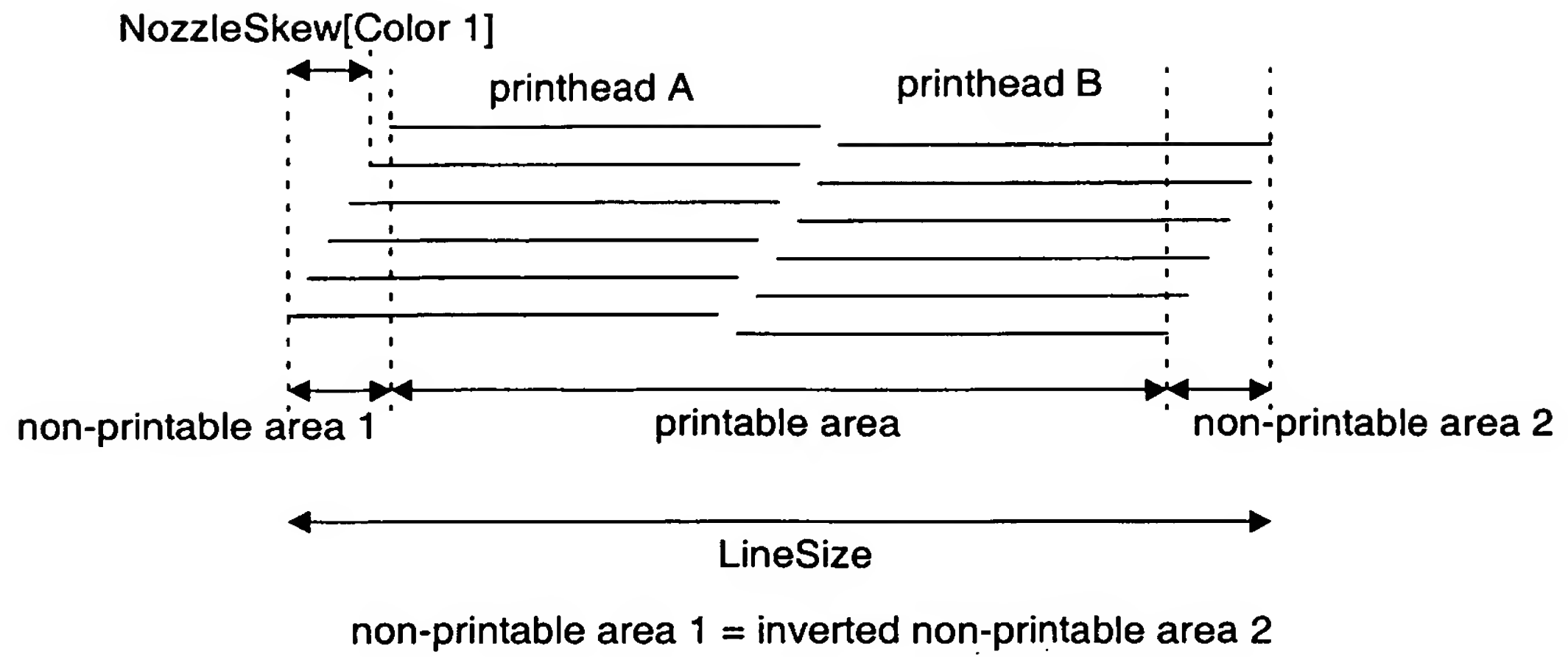


FIG. 257

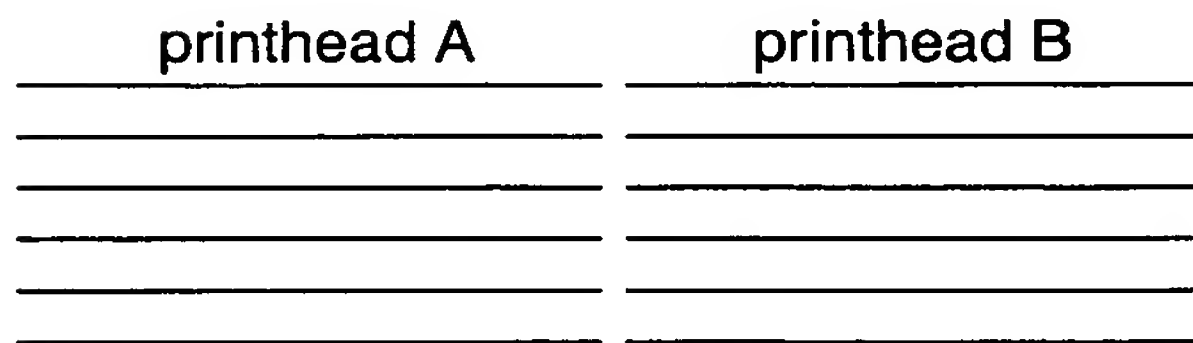


FIG. 258

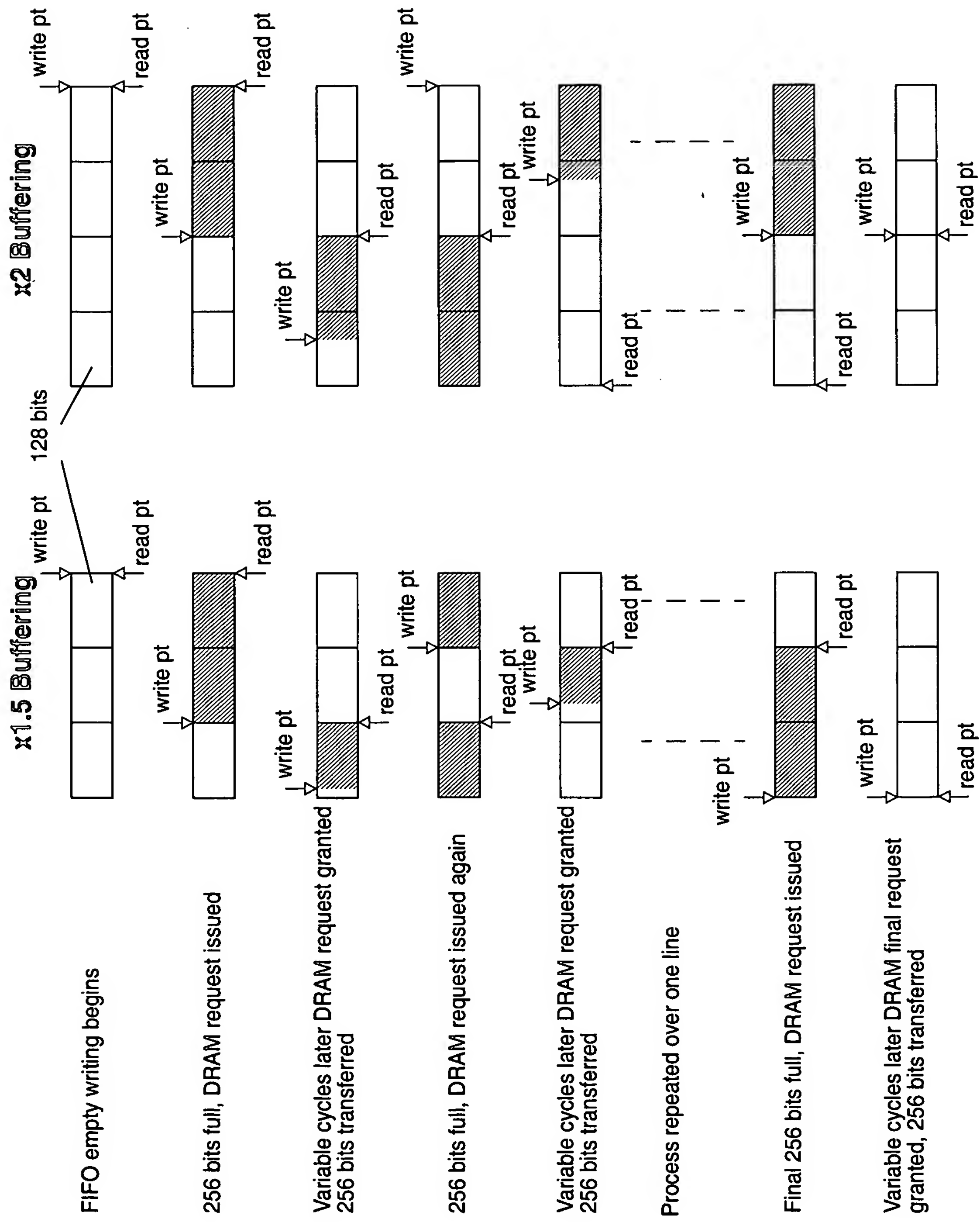


FIG. 259

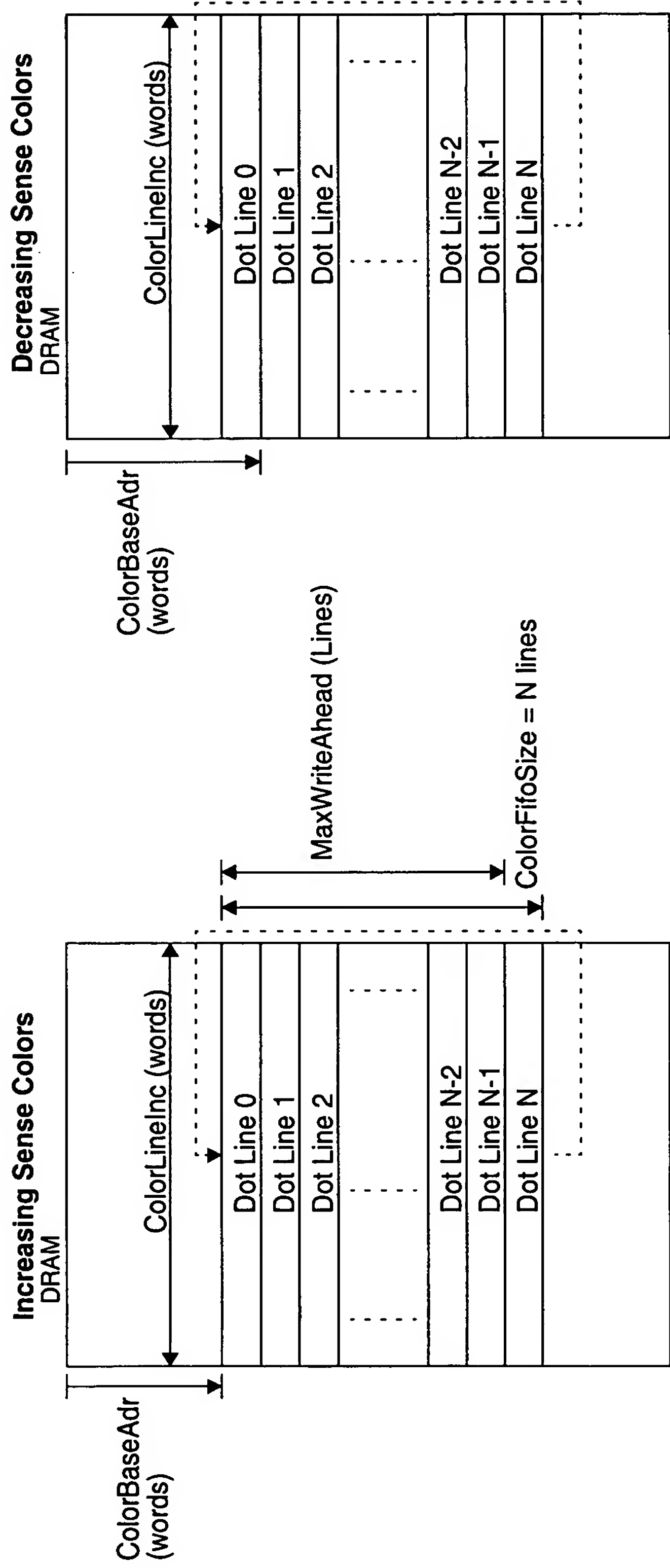


FIG. 262

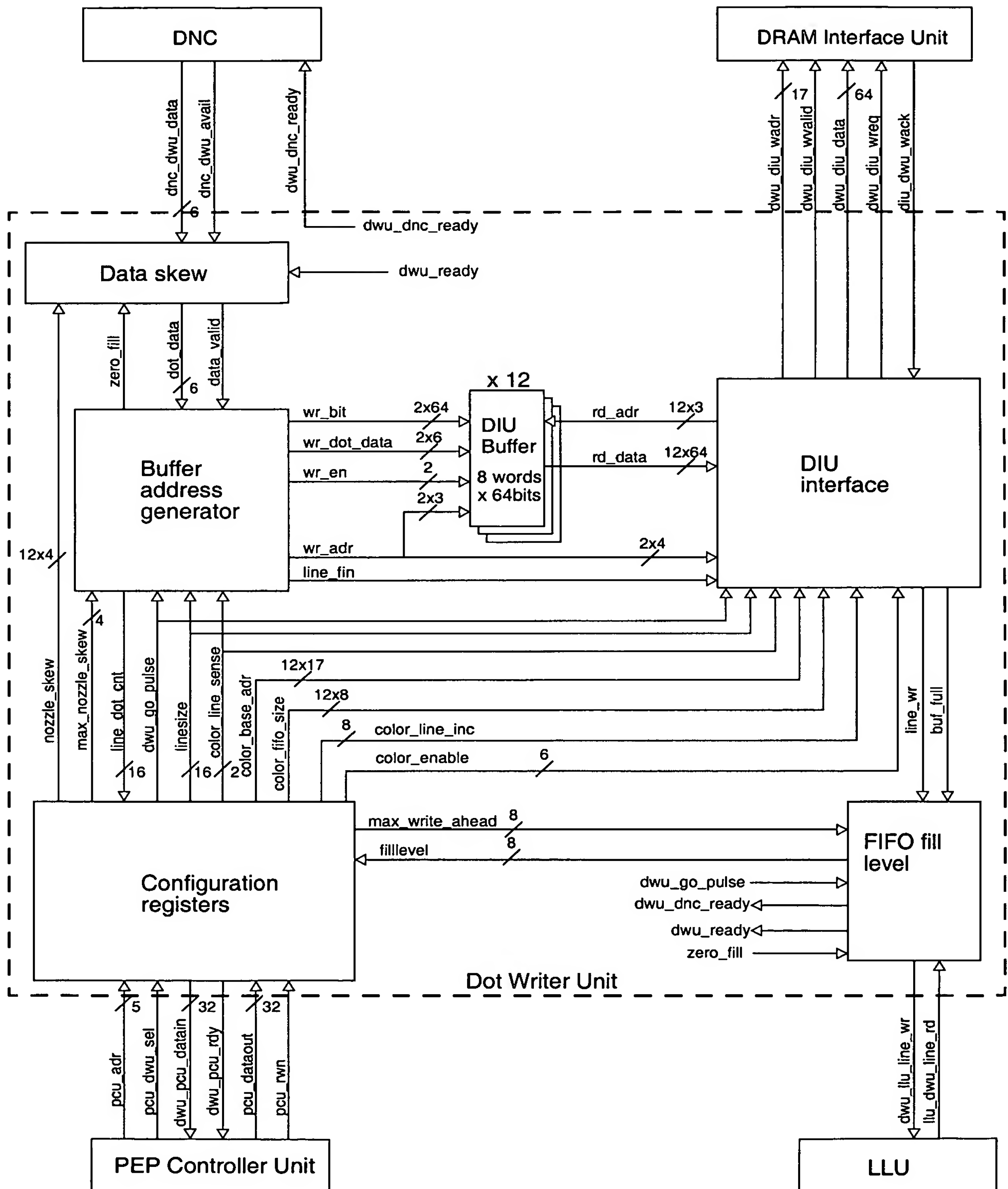


FIG. 263

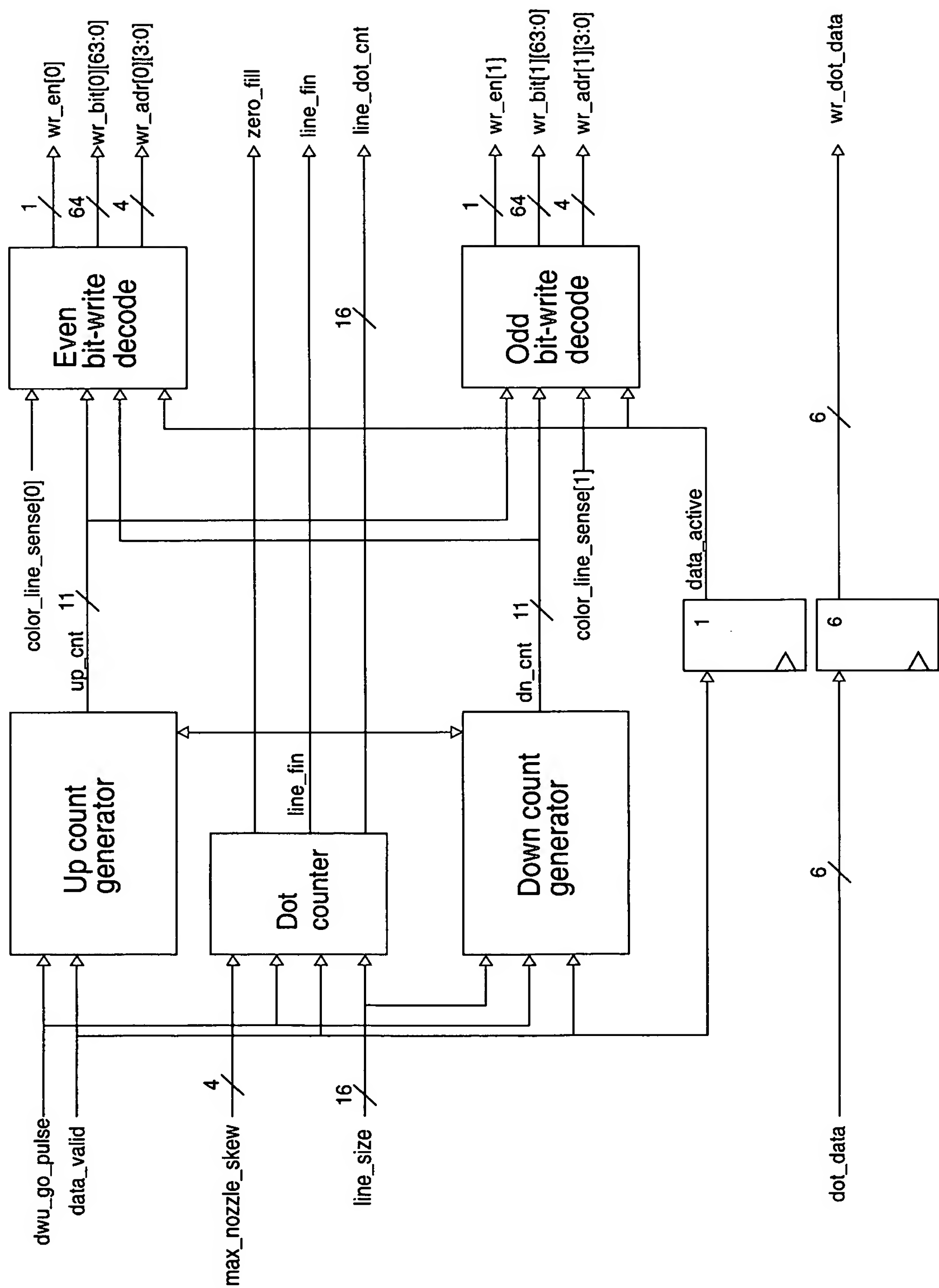


FIG. 264

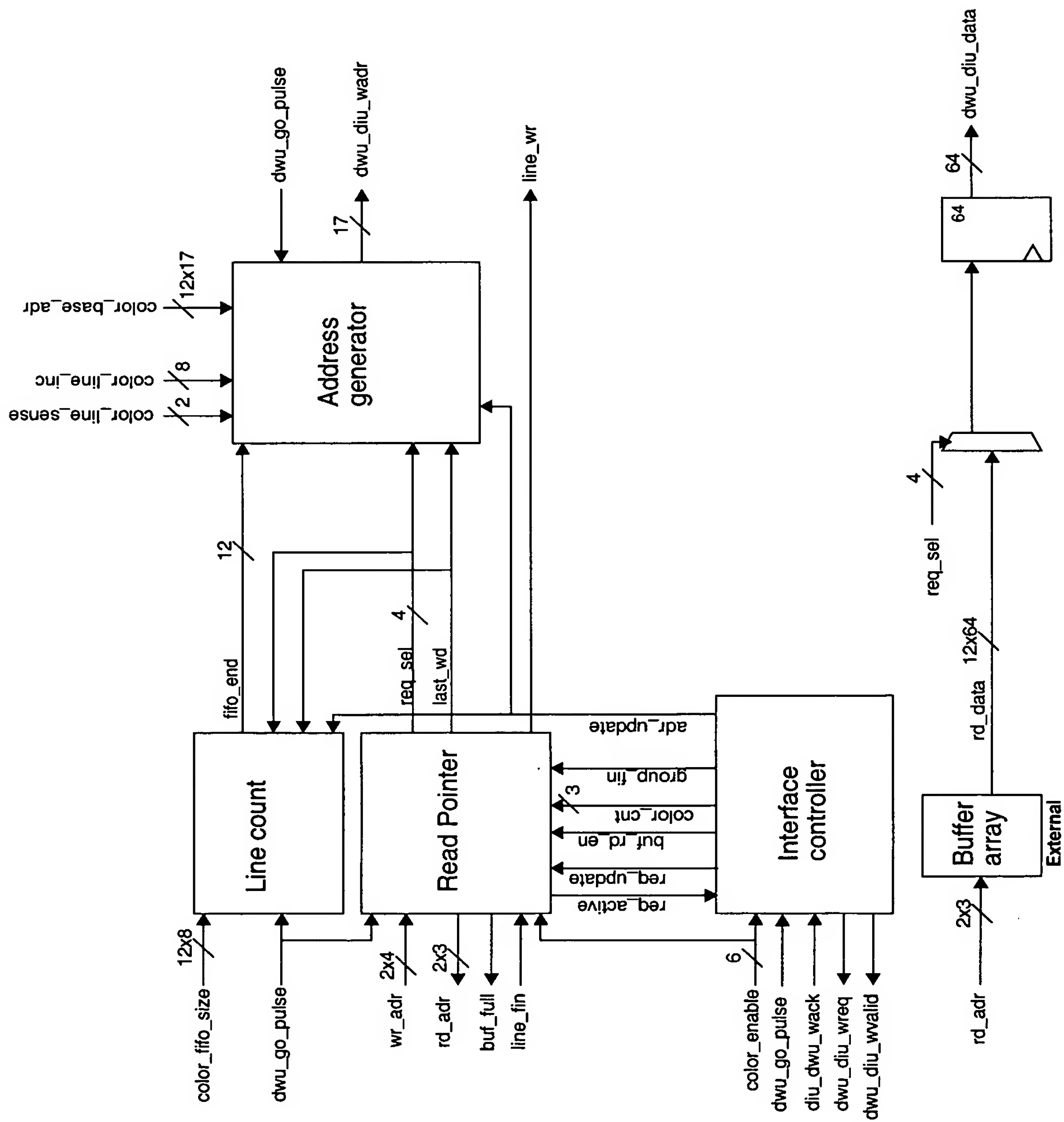


FIG. 265

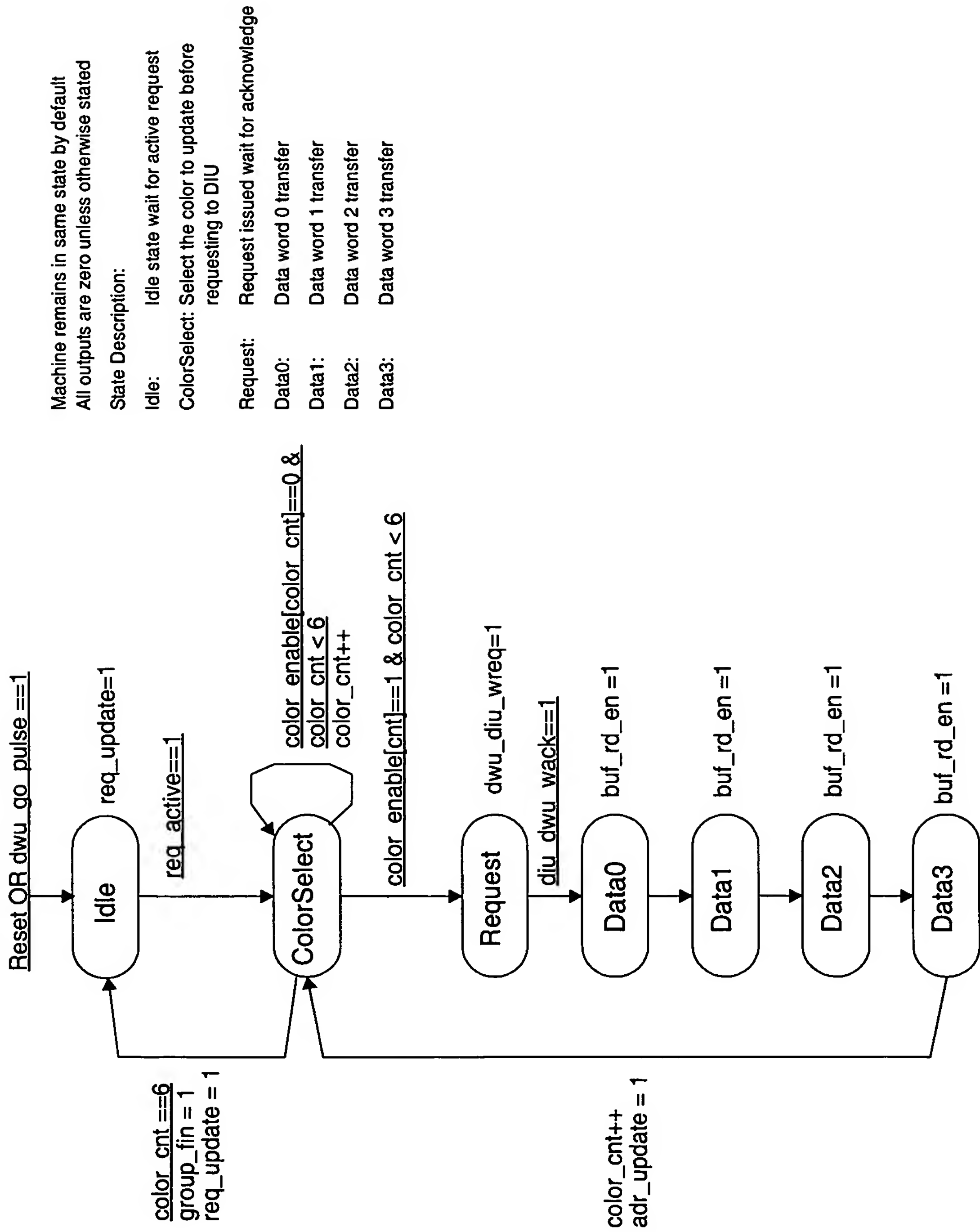


FIG. 266

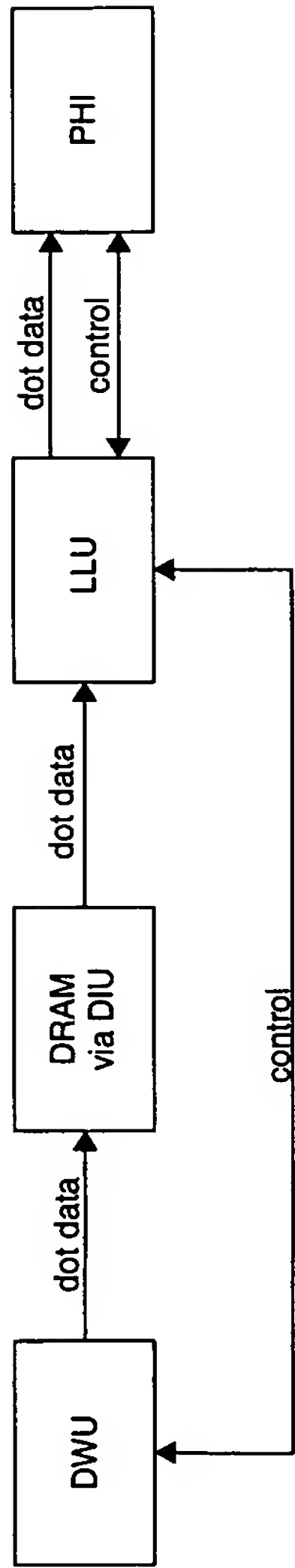


FIG. 267

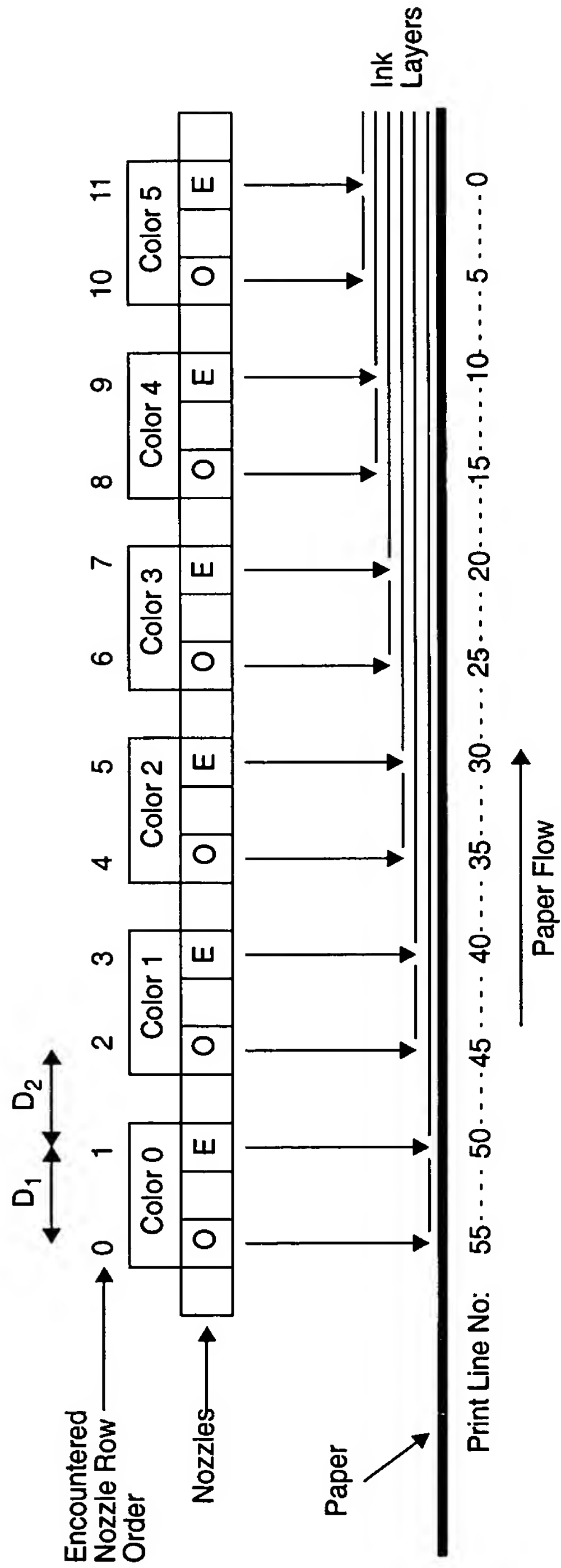
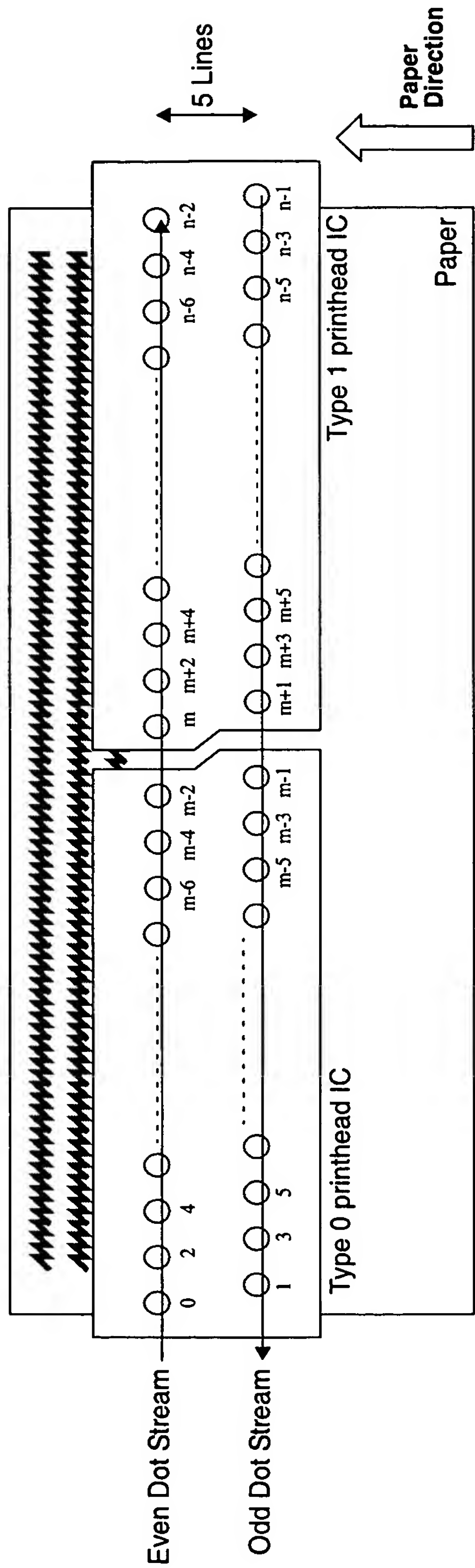


FIG. 268

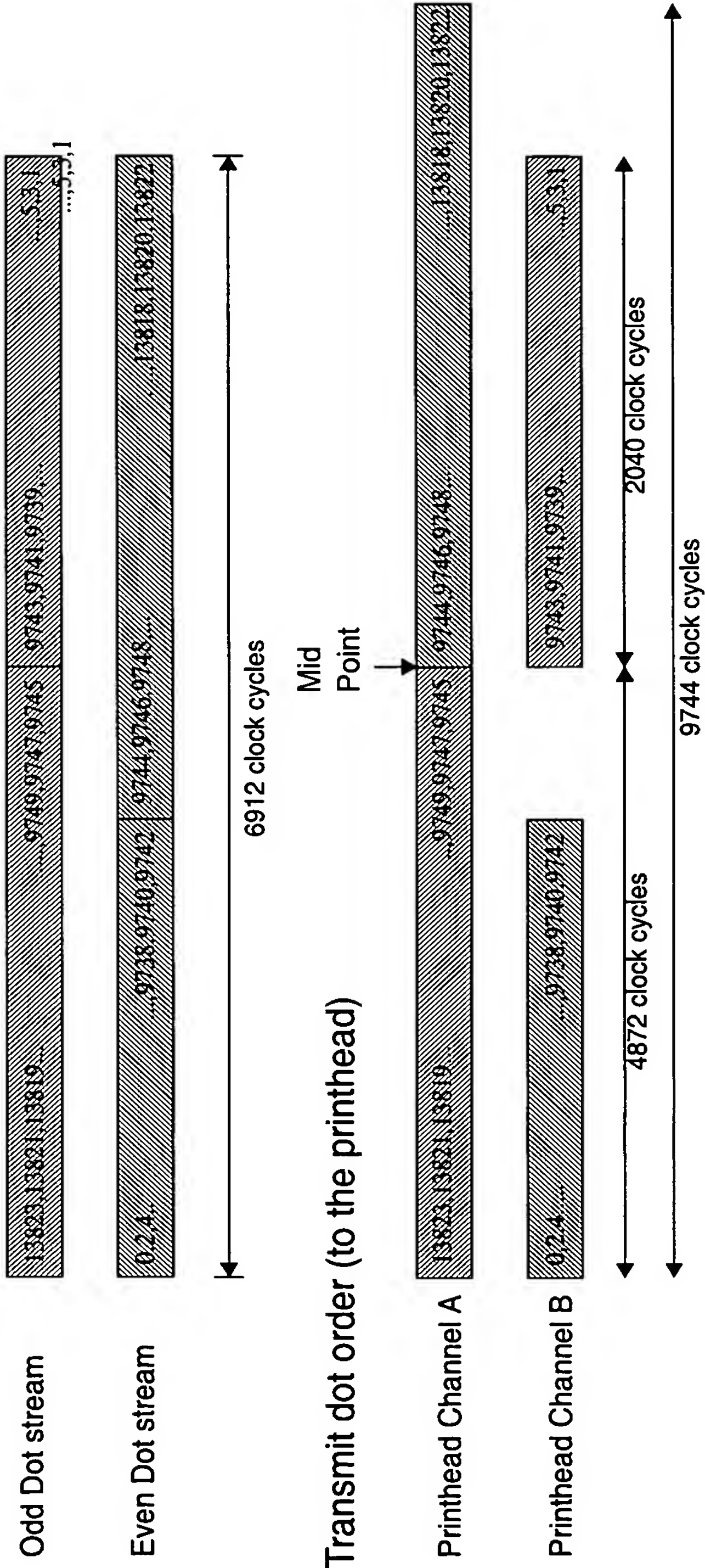


M - Midway point in dots
N - Number of dots in a line

Note: Paper passing under printhead

FIG. 269

Generate dot order (to the PHI)



- Even dots from Line Y
- Odd dots from Line Y-5

Example: Line with 13824 dots, with 7:3 printhead

FIG. 270

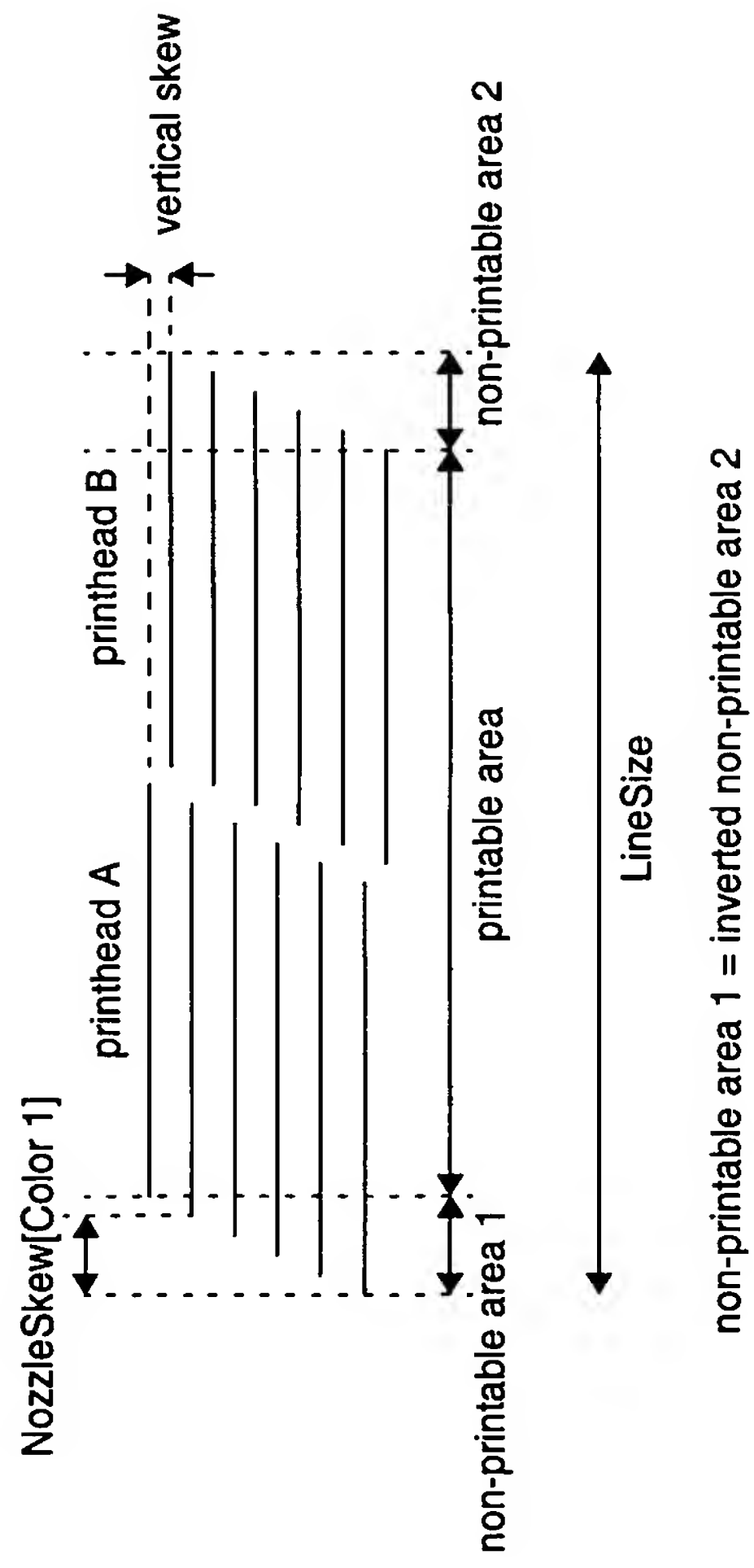


FIG. 271

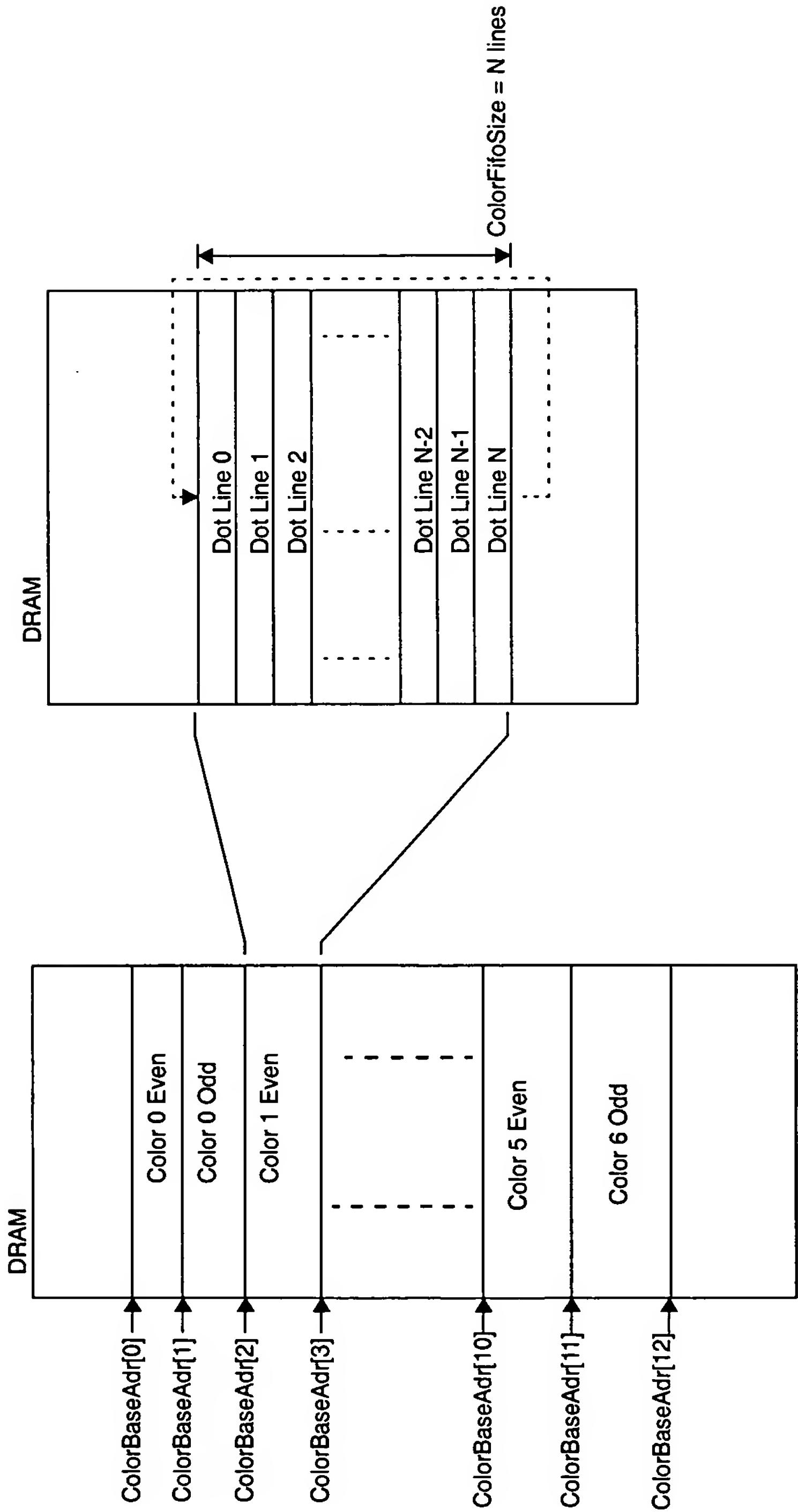


FIG. 272

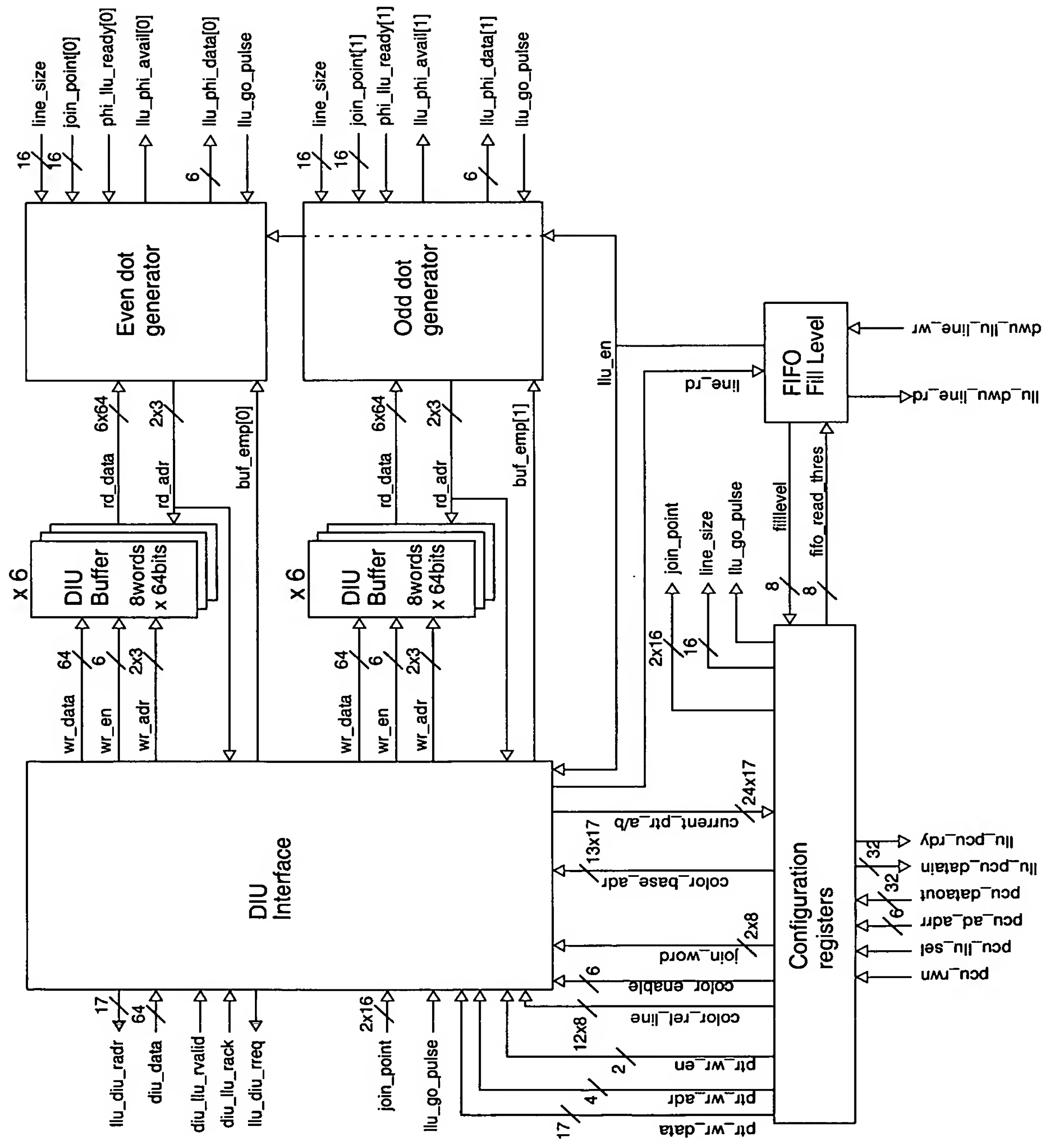


FIG. 273

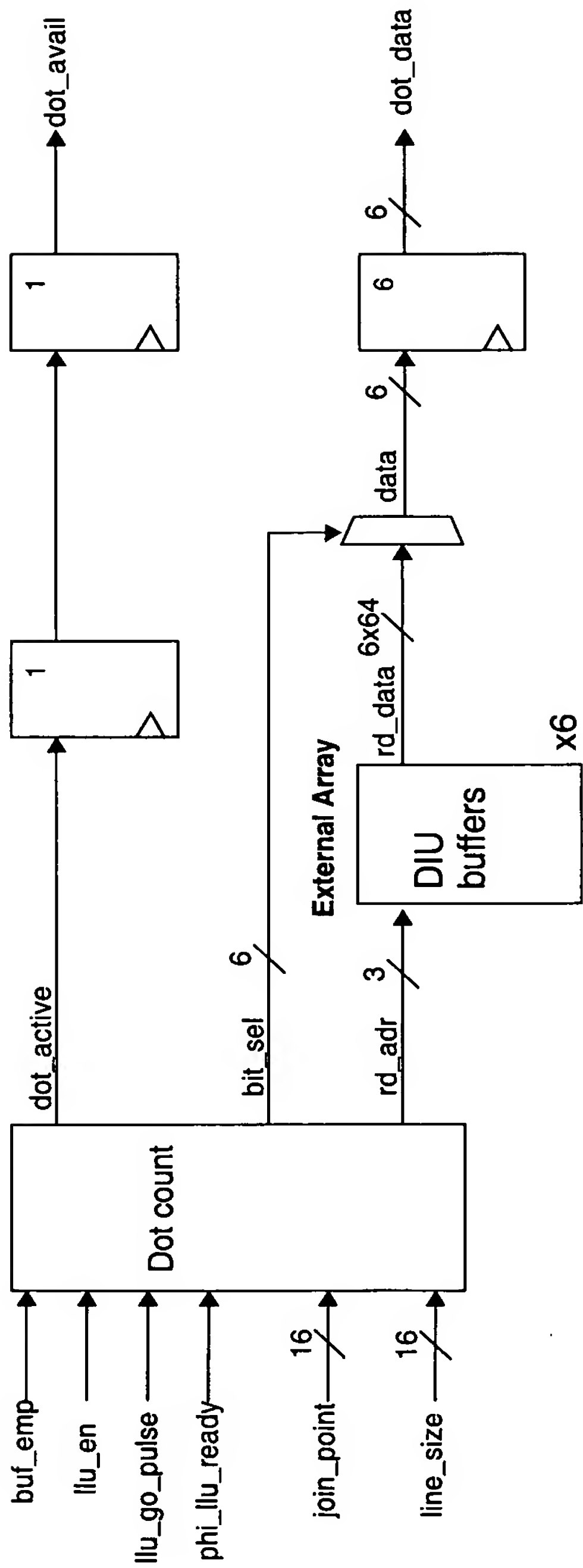


FIG. 274

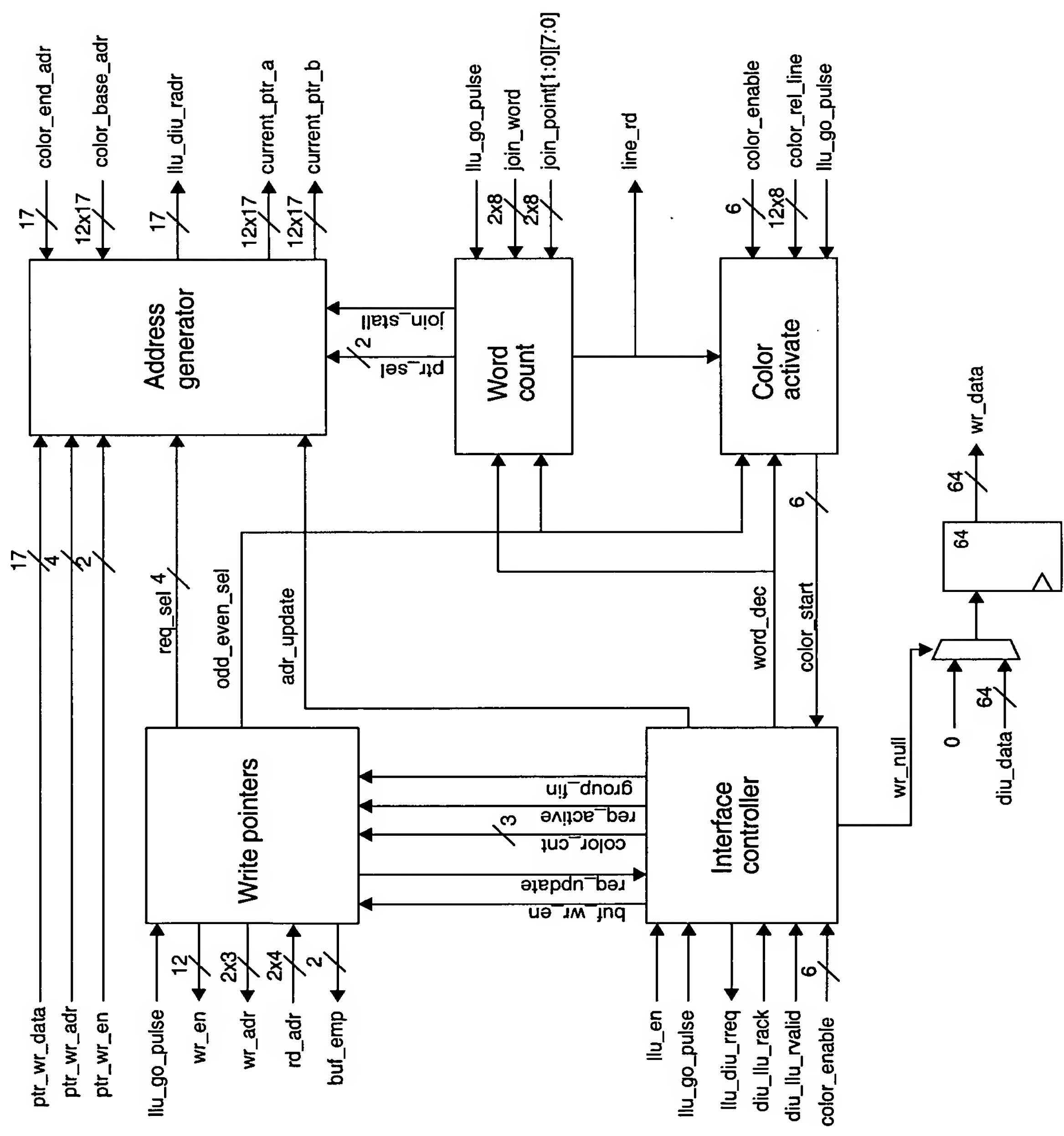


FIG. 275

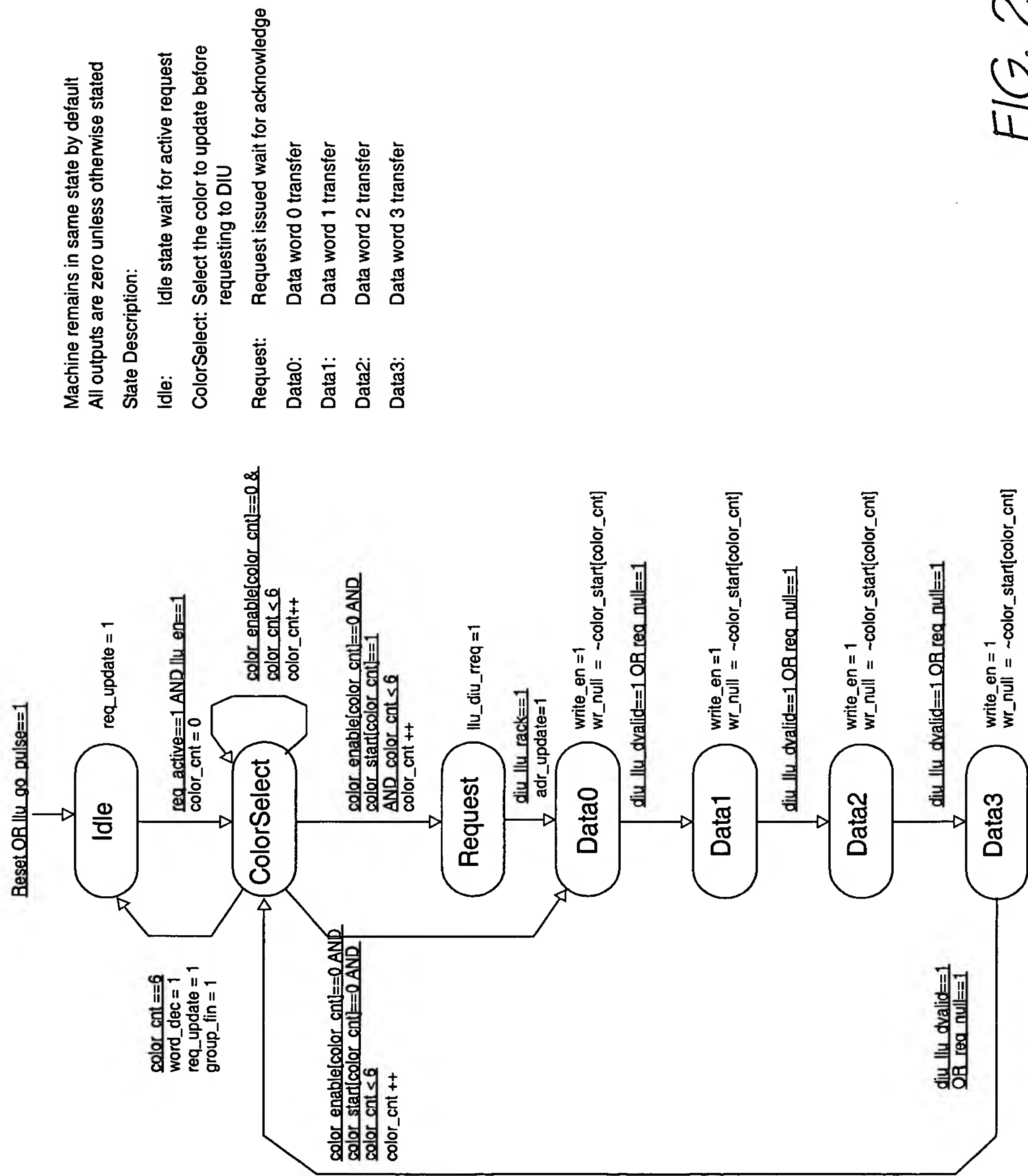


FIG. 276

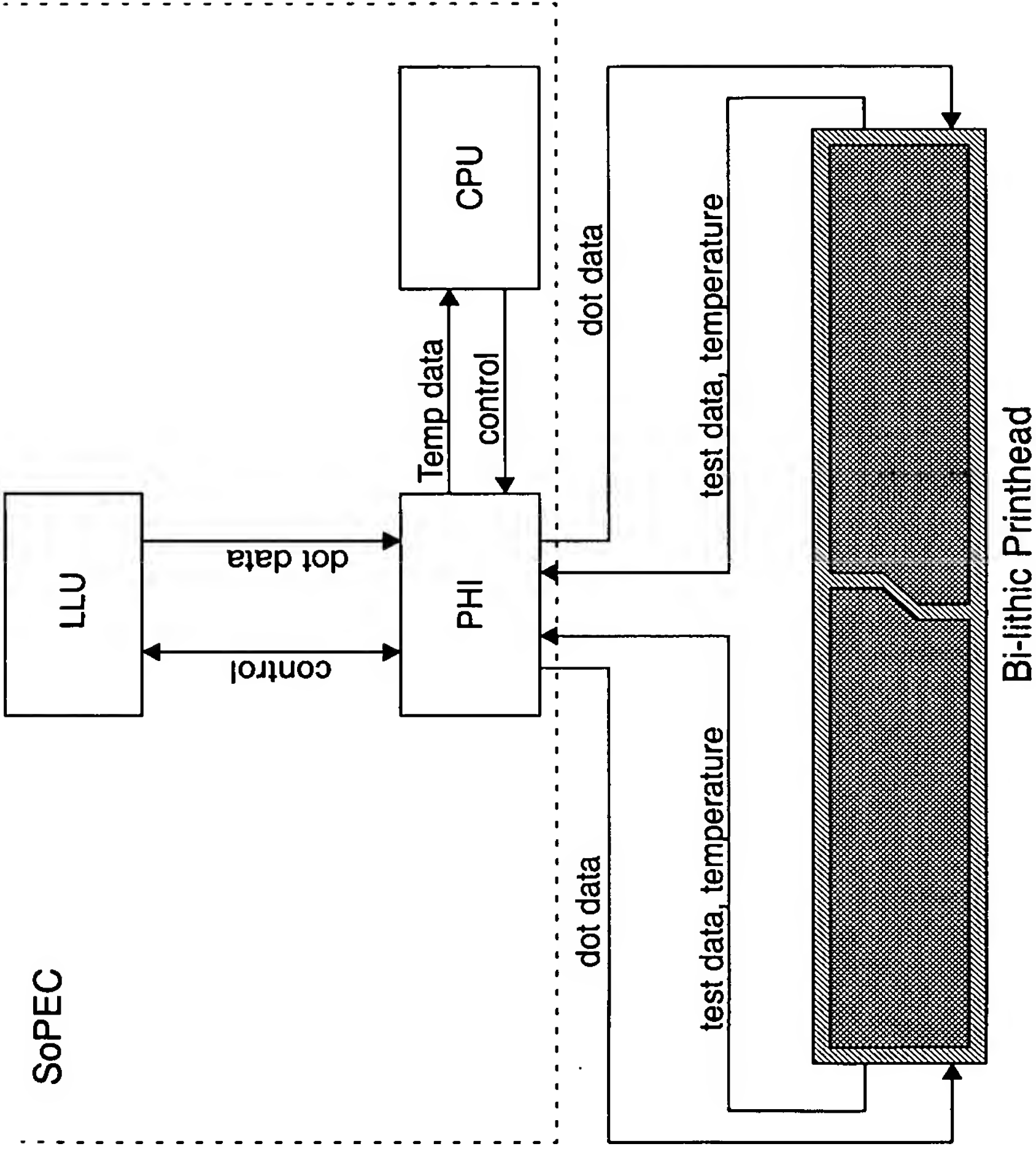


FIG. 277

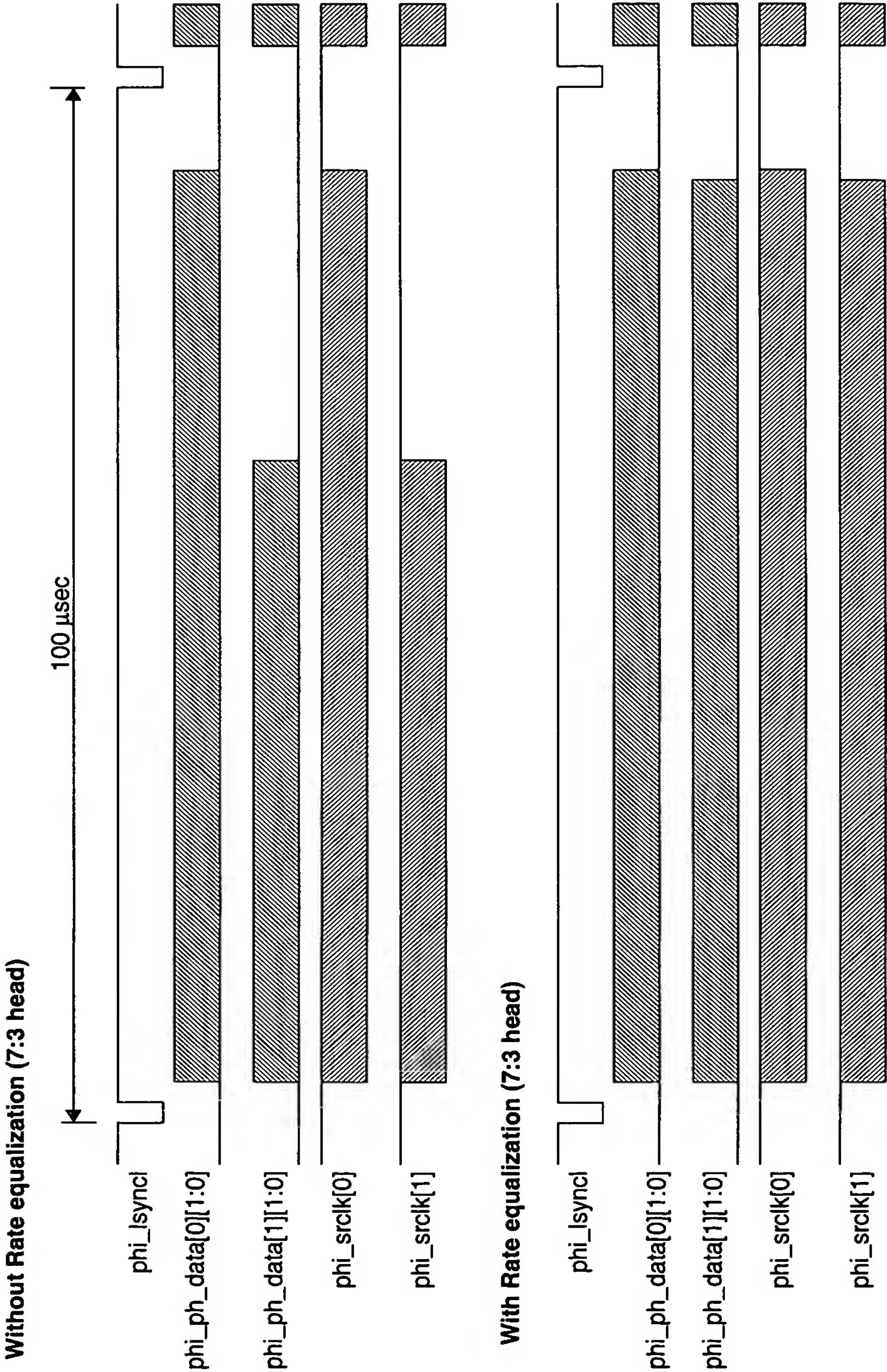
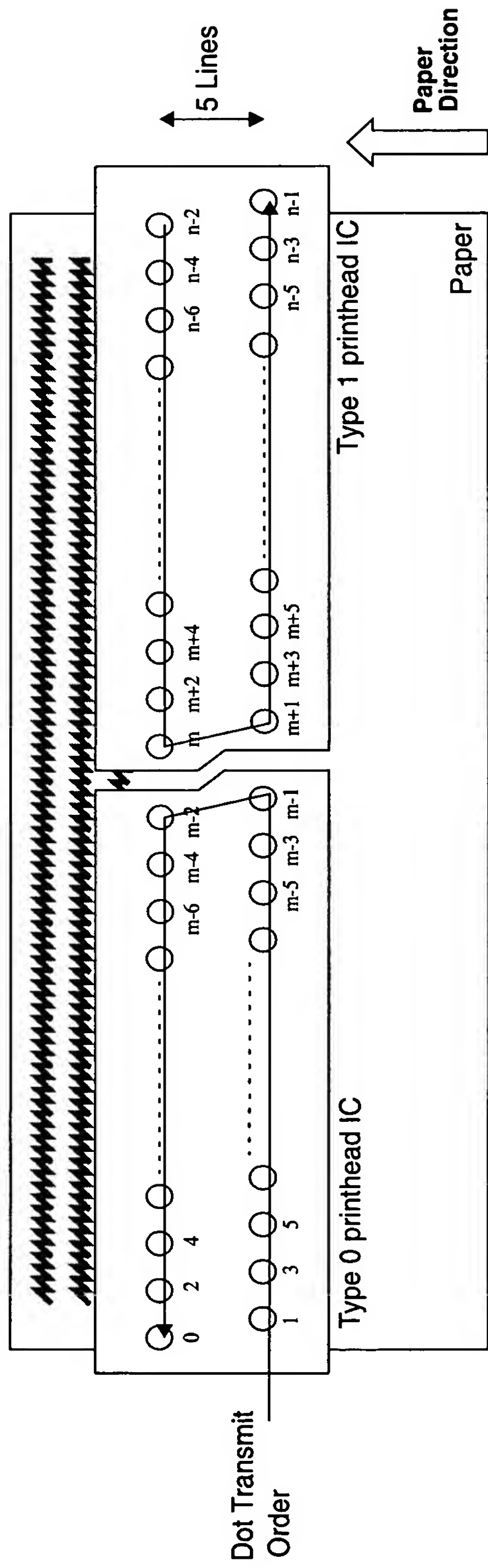


FIG. 279

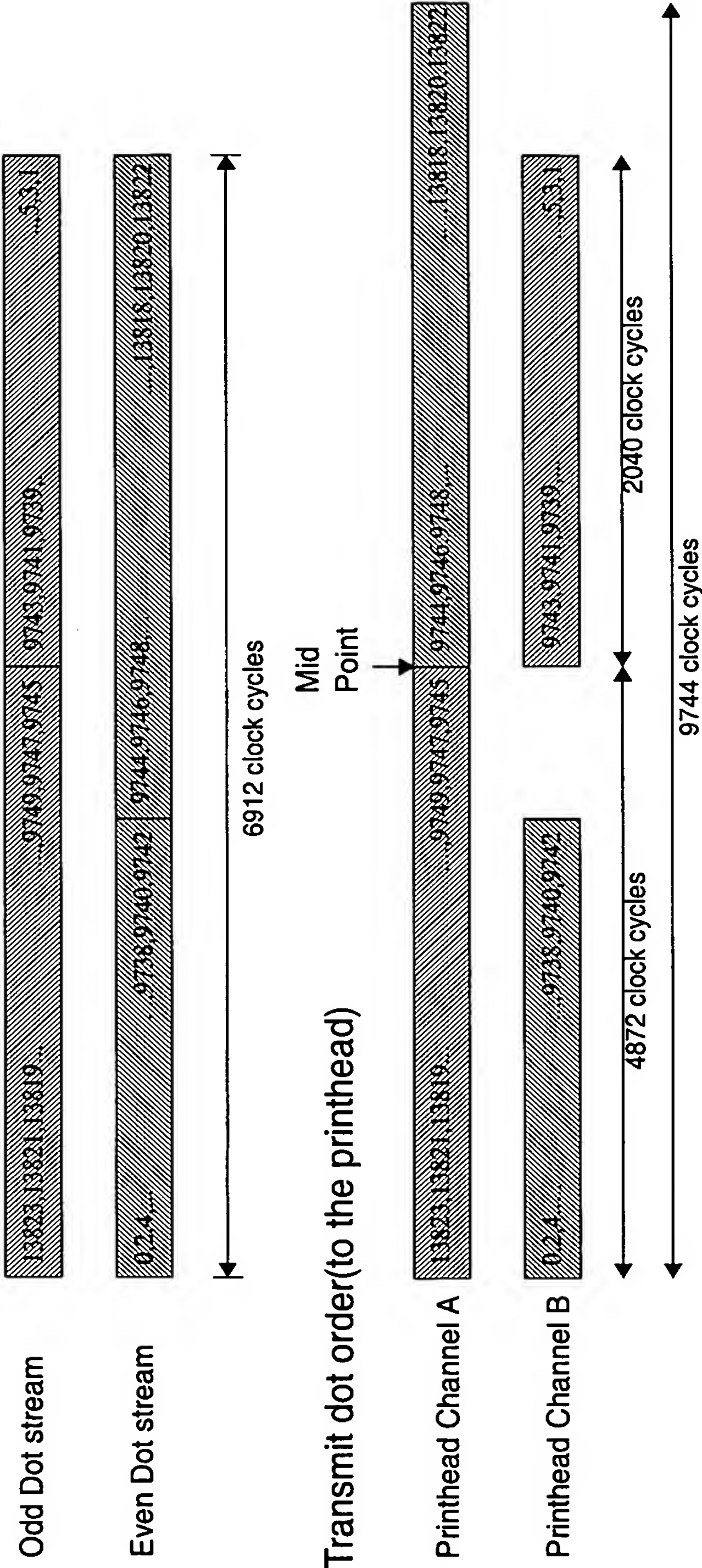


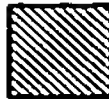
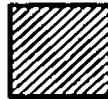
M - Midway point in dots
N - Number of dots in a line

Note: Paper passing under printhead

FIG. 280

Generate dot order (from the LLU)

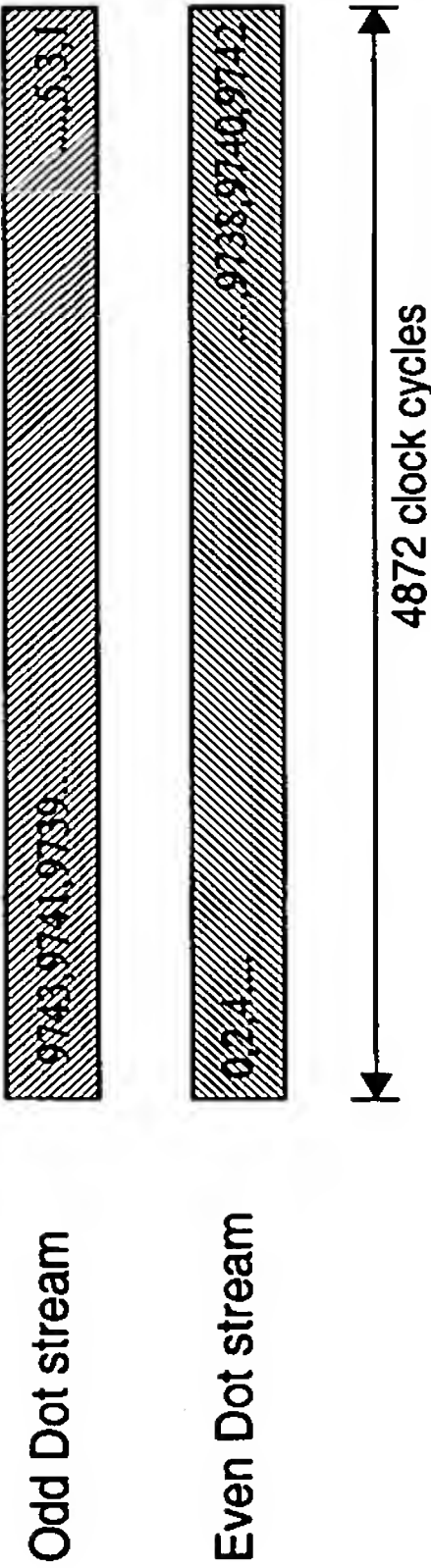


-  Even dots from Line Y
-  Odd dots from Line Y-5

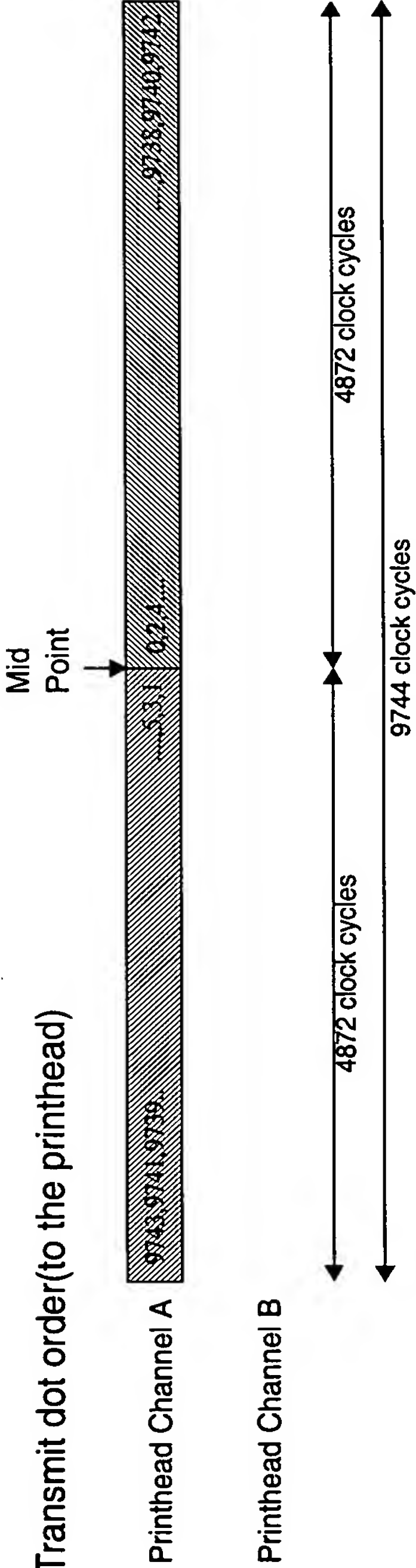
Example: Line with 13824 dots, with 7:3 printhead



FIG. 281

Generate dot order (from the LLU)



Transmit dot order(to the printhead)



-  Even dots from Line Y
-  Odd dots from Line Y-5

Example: Line with 9744 dots, with 7:0 printhead

FIG. 282

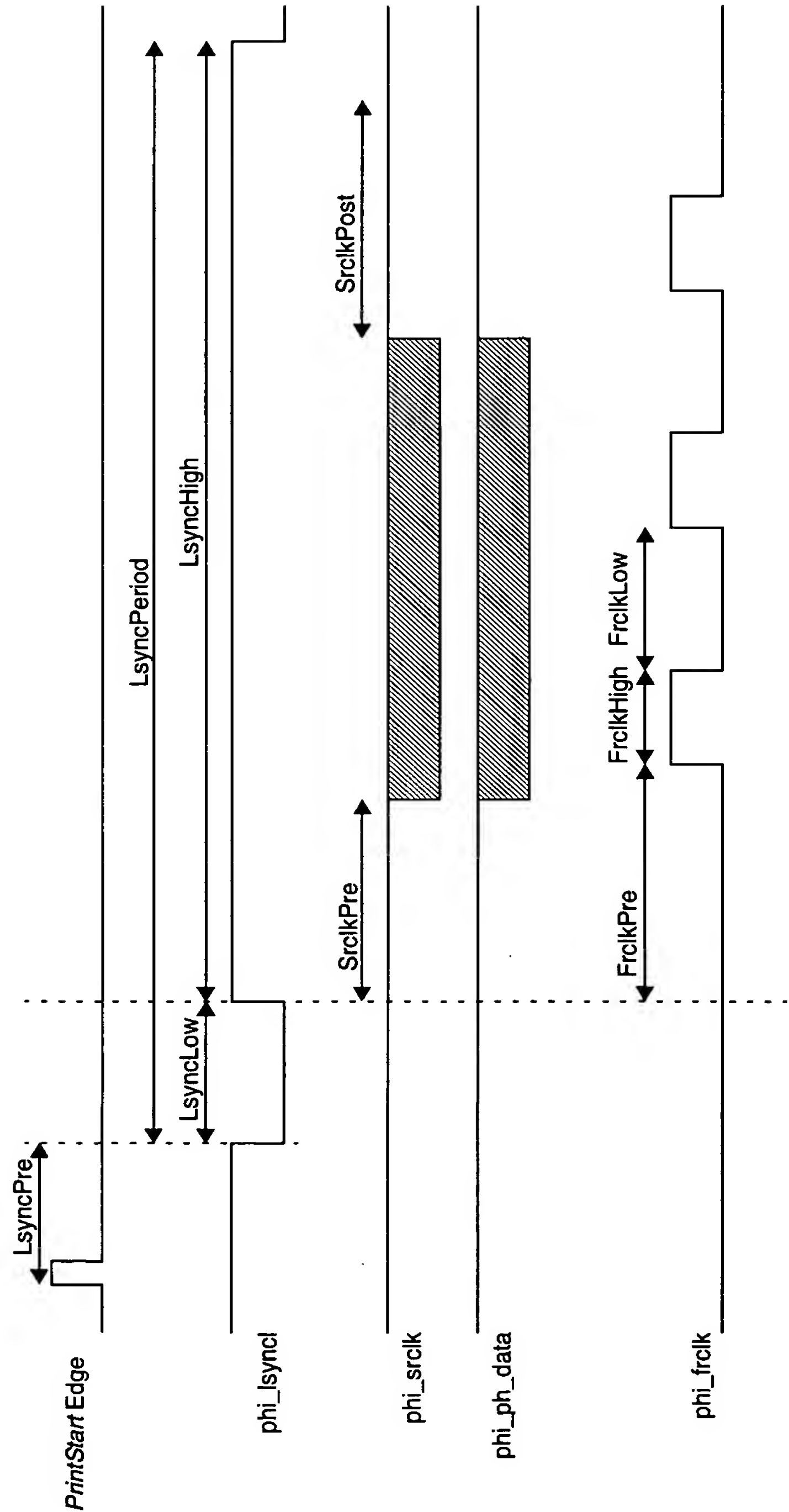


FIG. 283

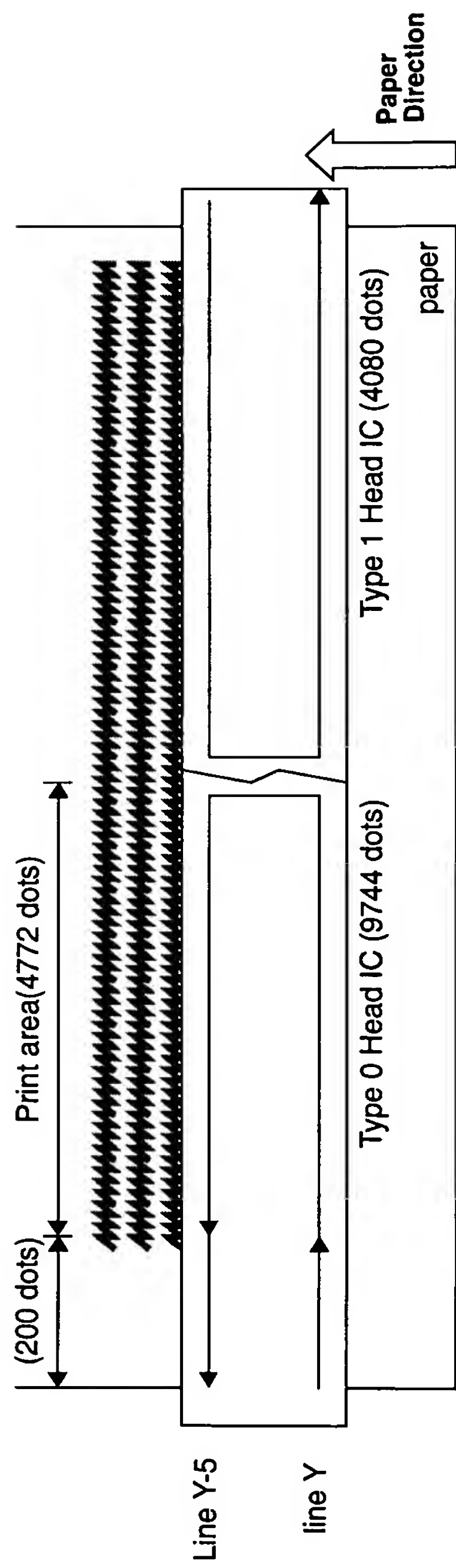


FIG. 284

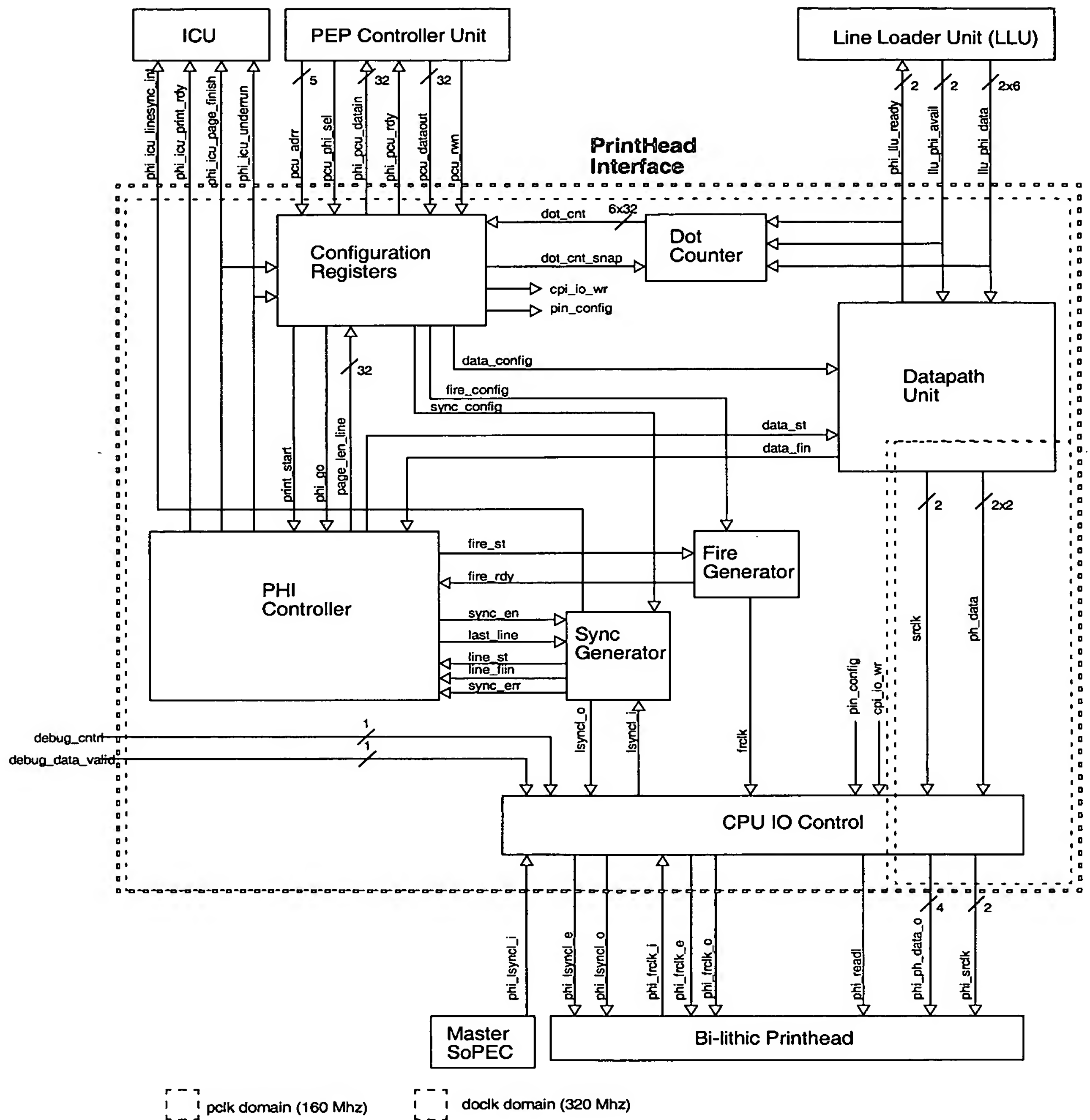


FIG. 285

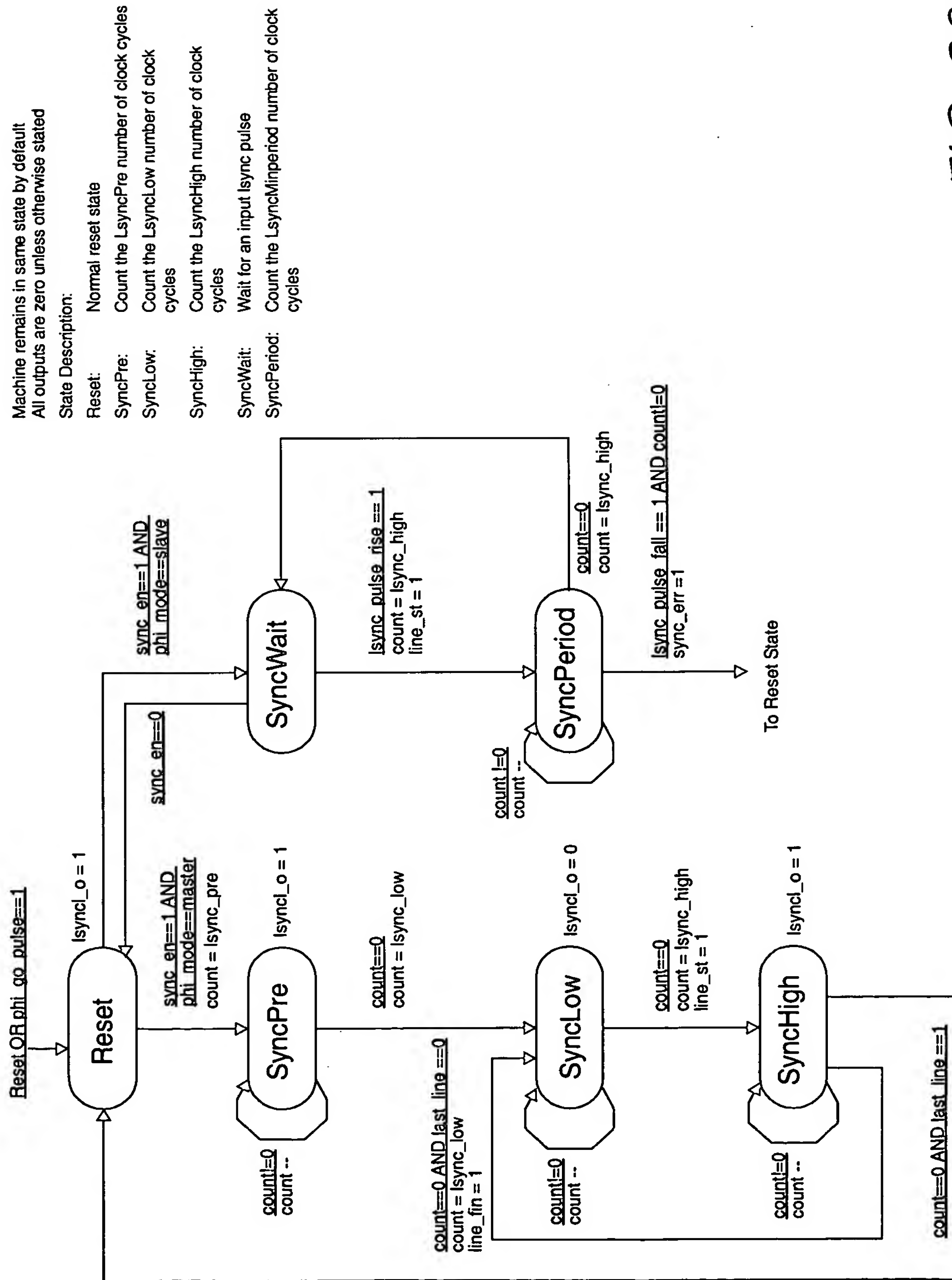


FIG. 286

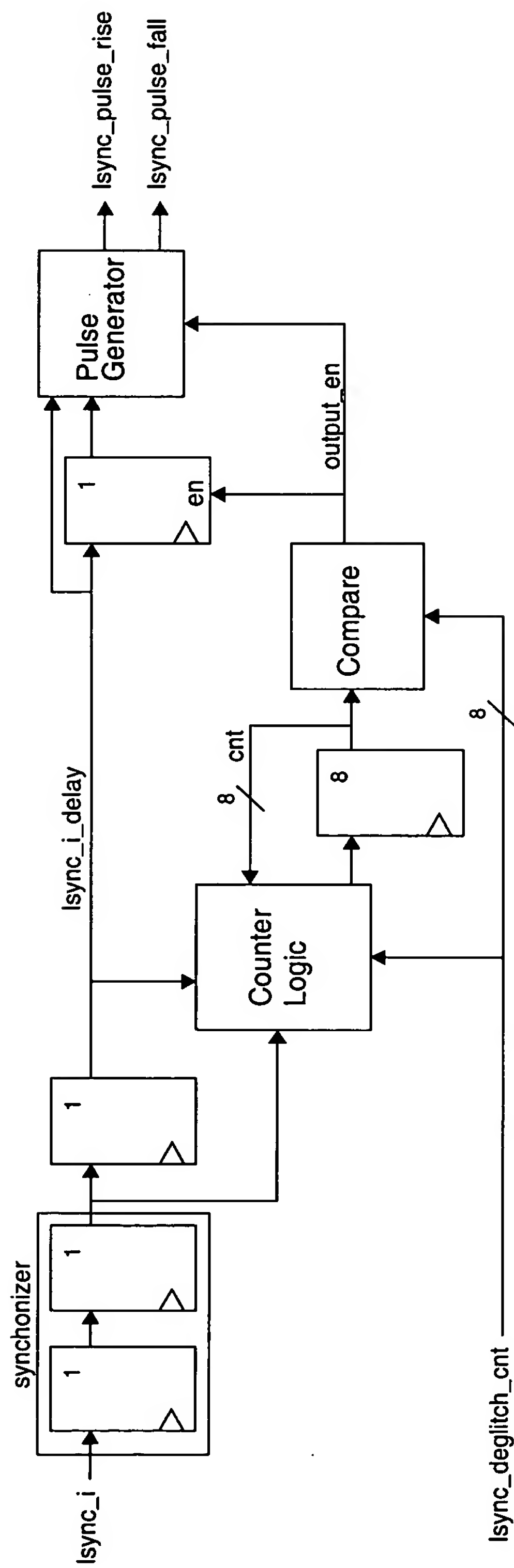


FIG. 287

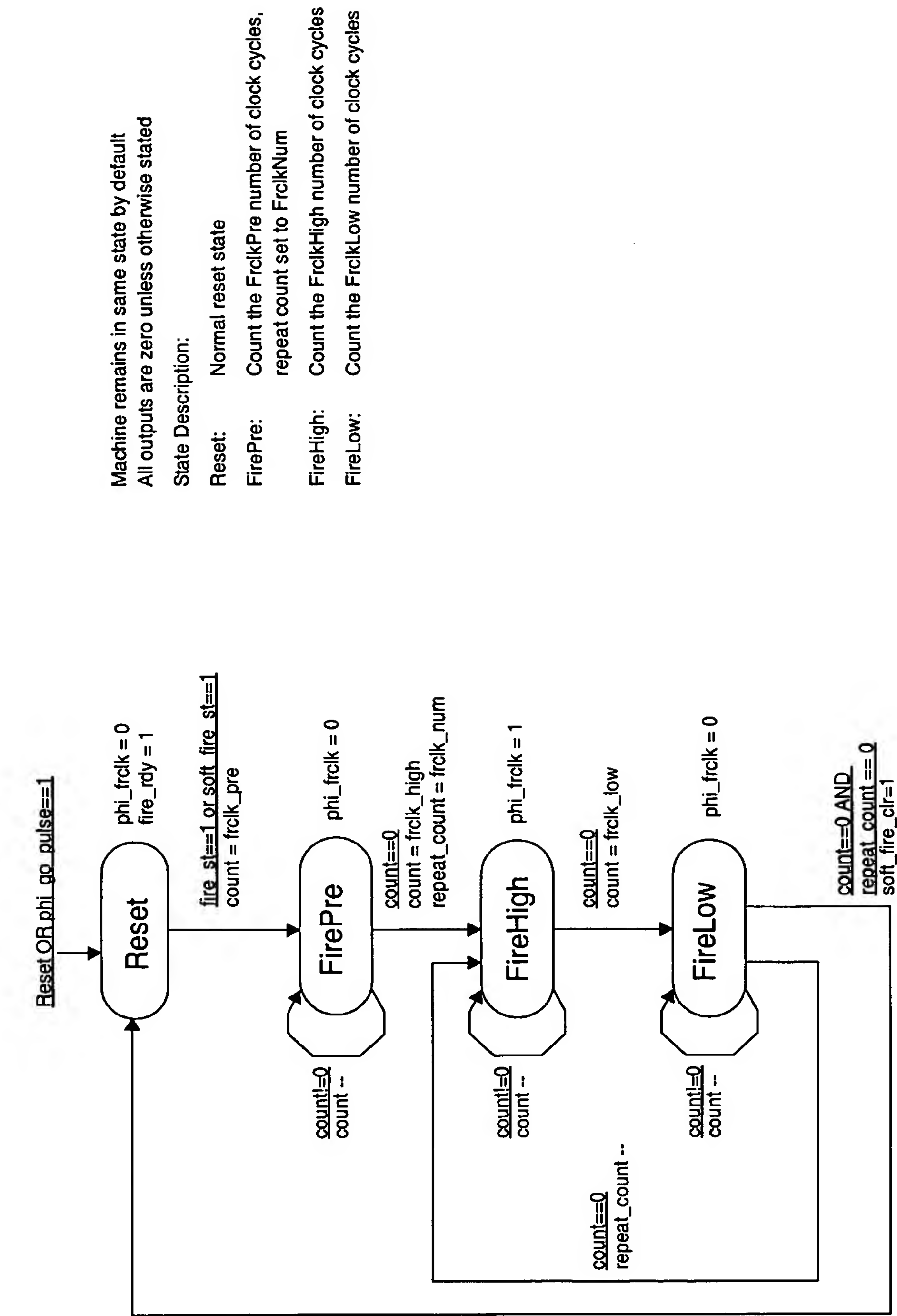


FIG. 288

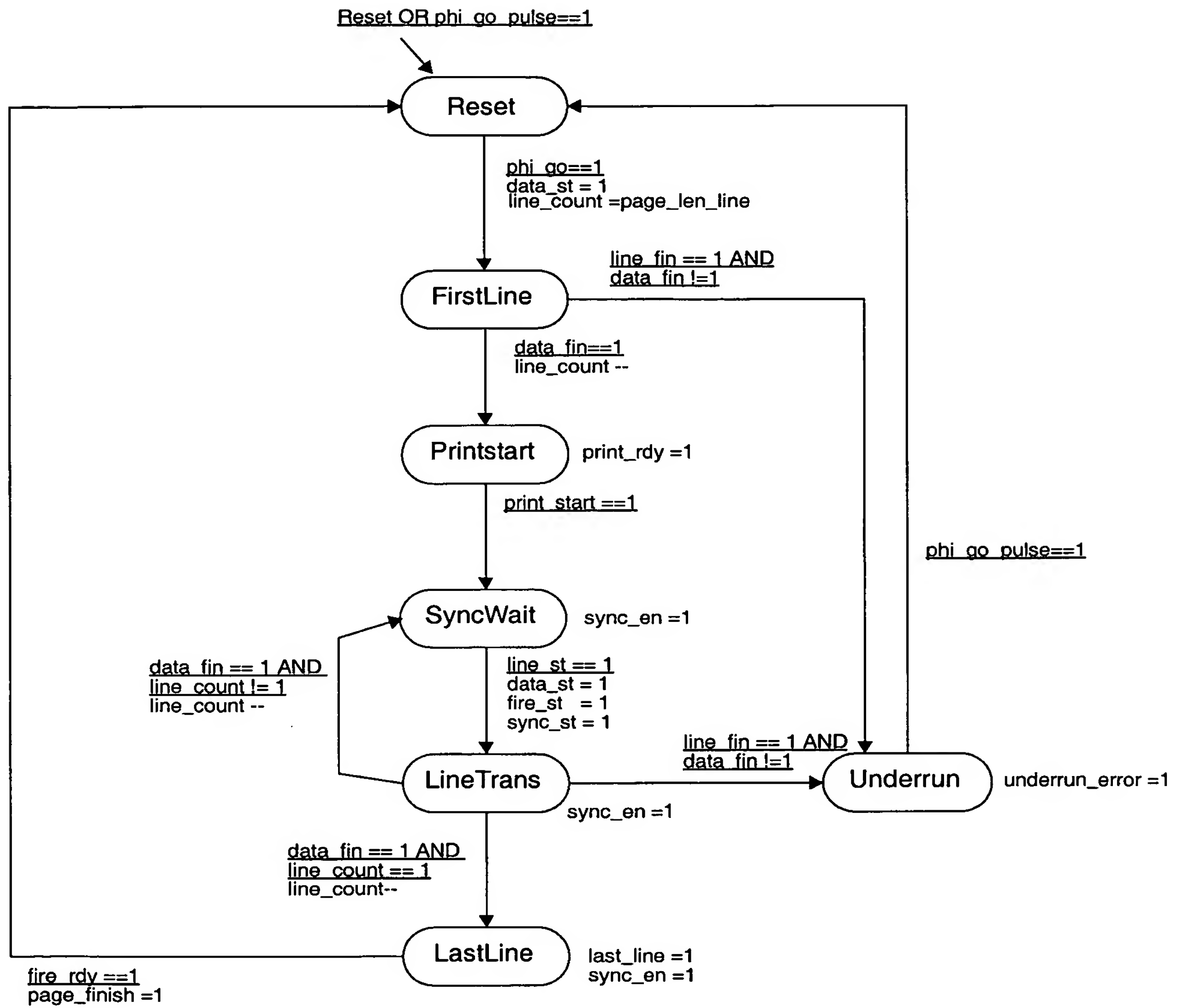


FIG. 289

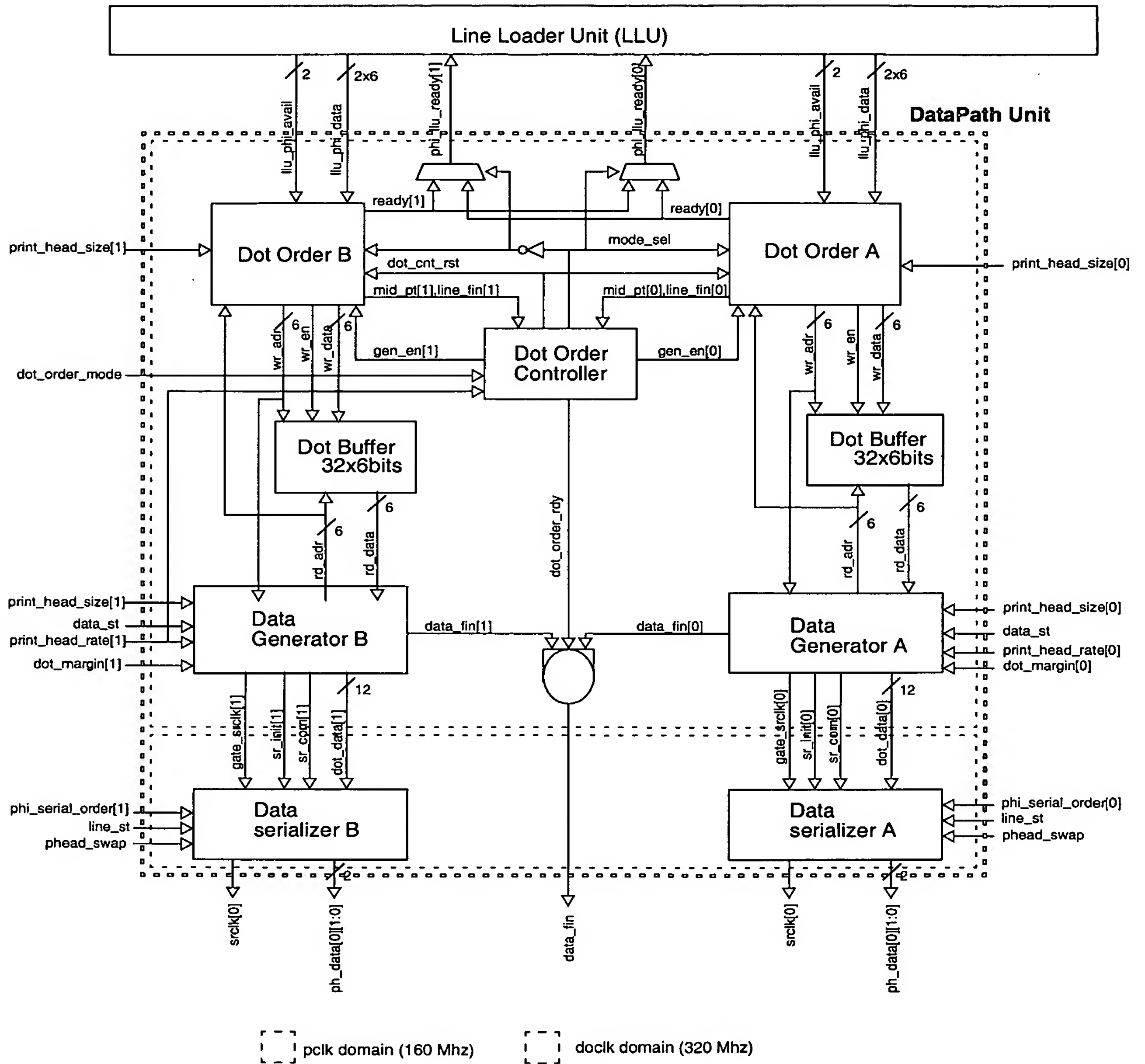
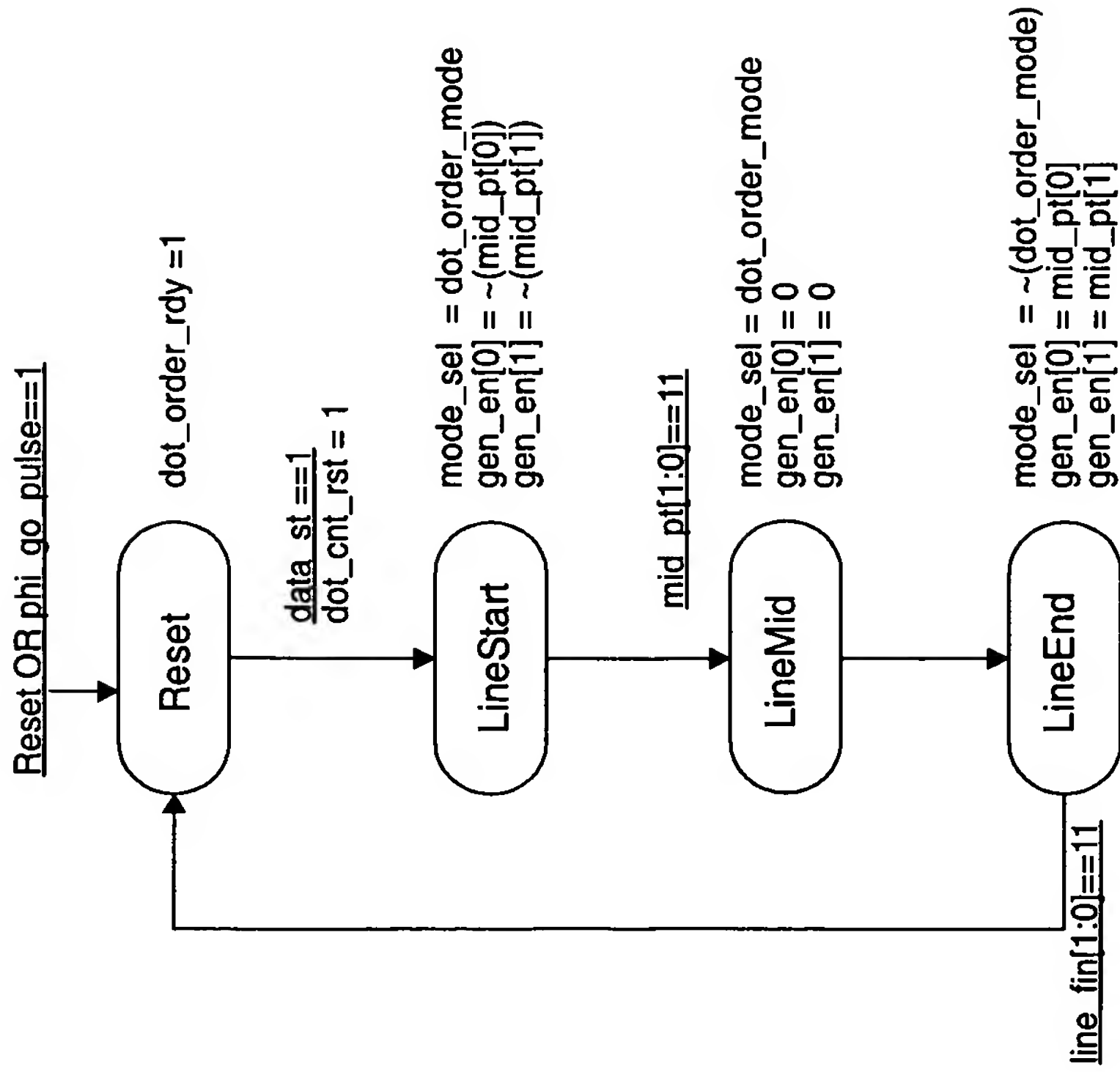


FIG. 290



Machine remains in same state by default
All outputs are zero unless otherwise stated

State Description:

Reset: Normal reset state

LineStart: Start processing first part of the line, wait for both mid_pt to be active

LineMid: Switch over wait state allow pipeline to clear

LineEnd: Line end processing wait for both line_fin to be active

FIG. 291

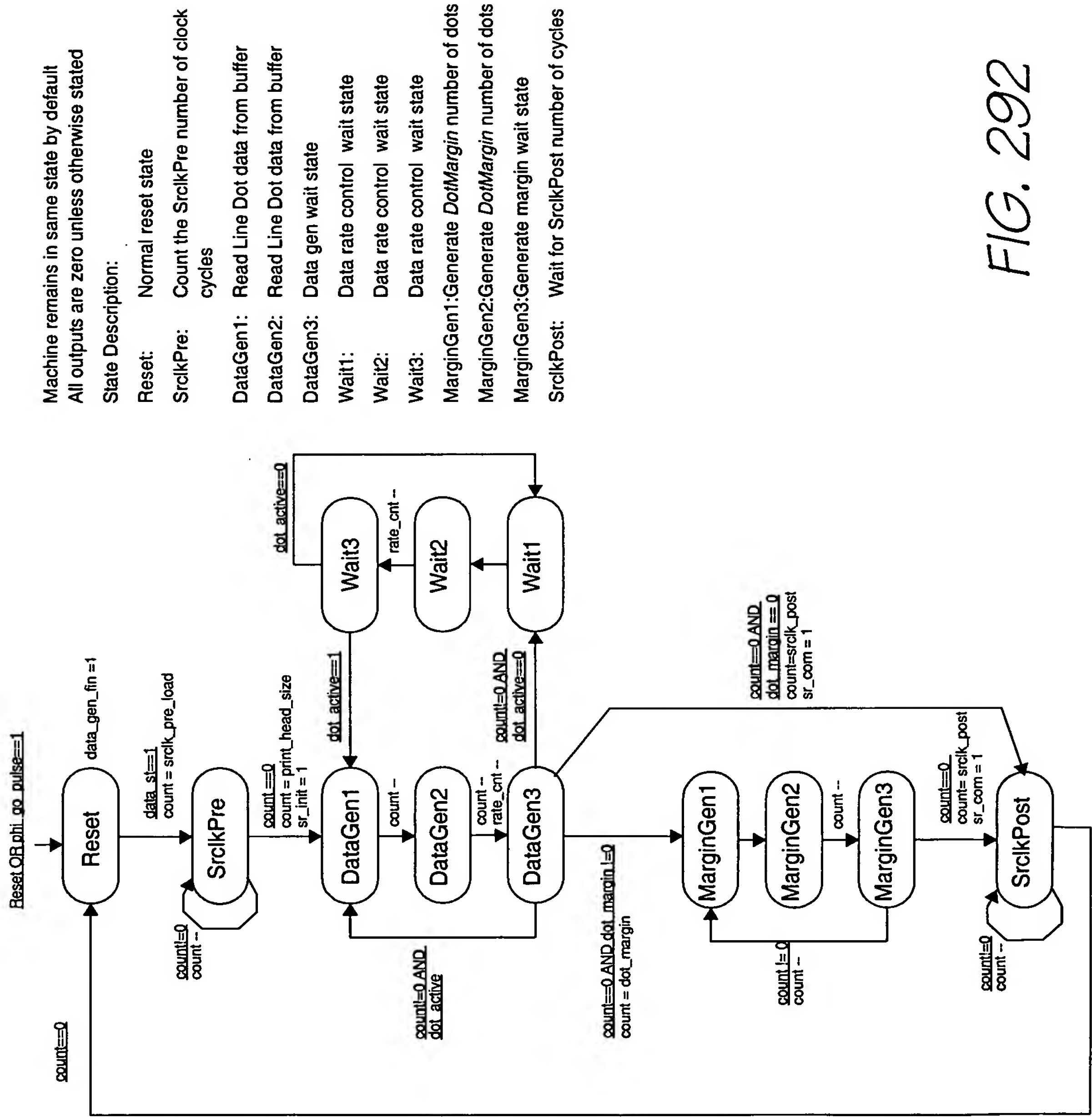


FIG. 292

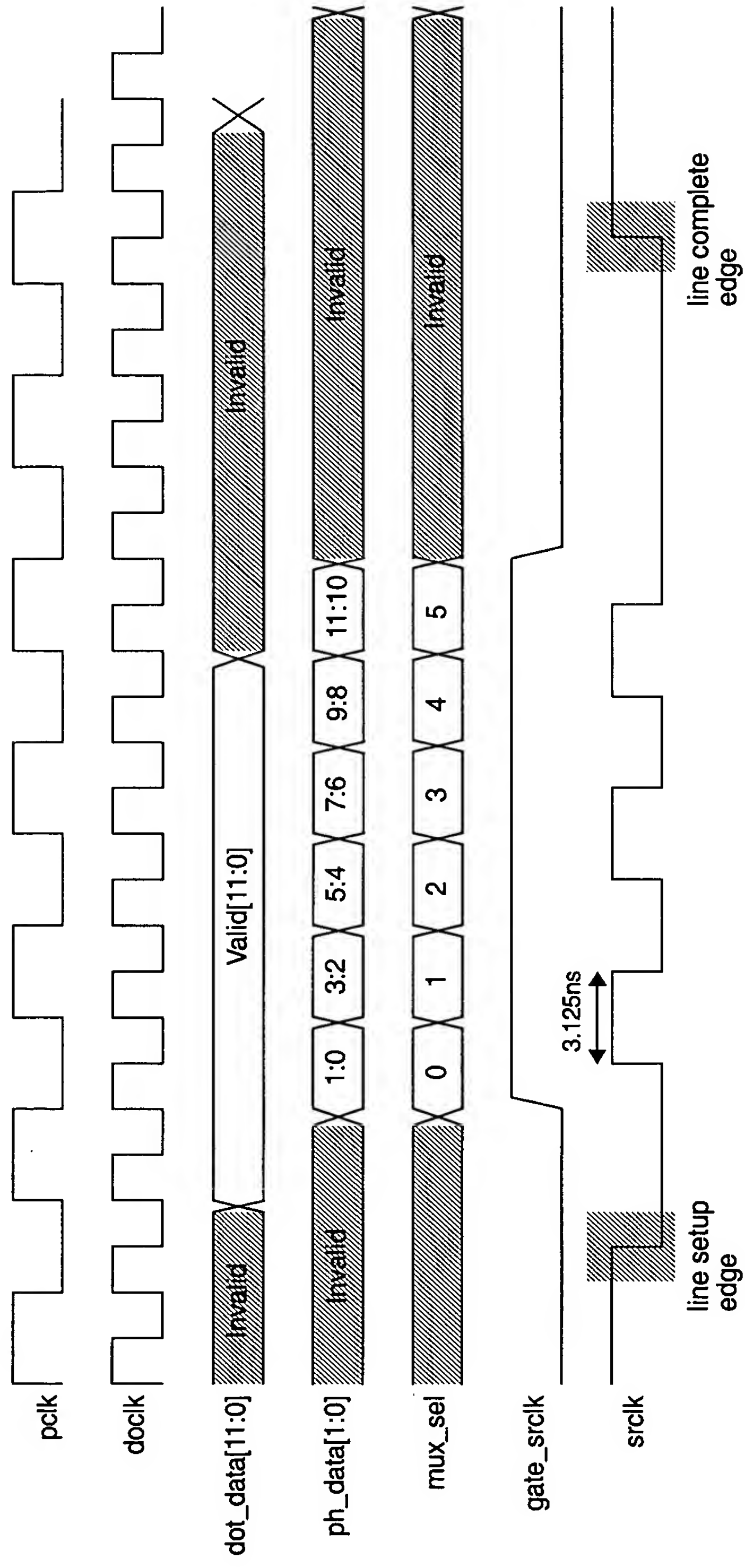


FIG. 293

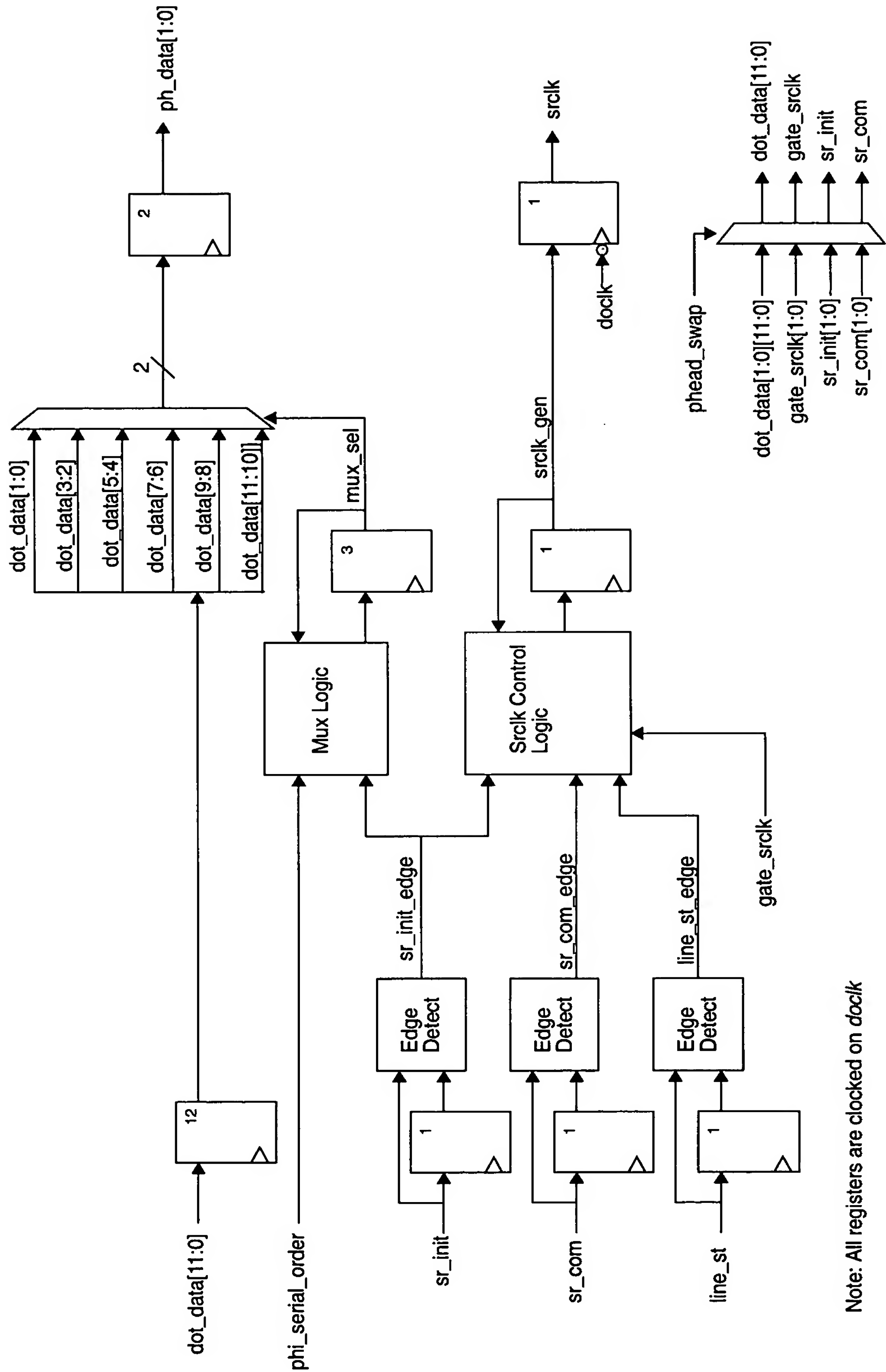


FIG. 294

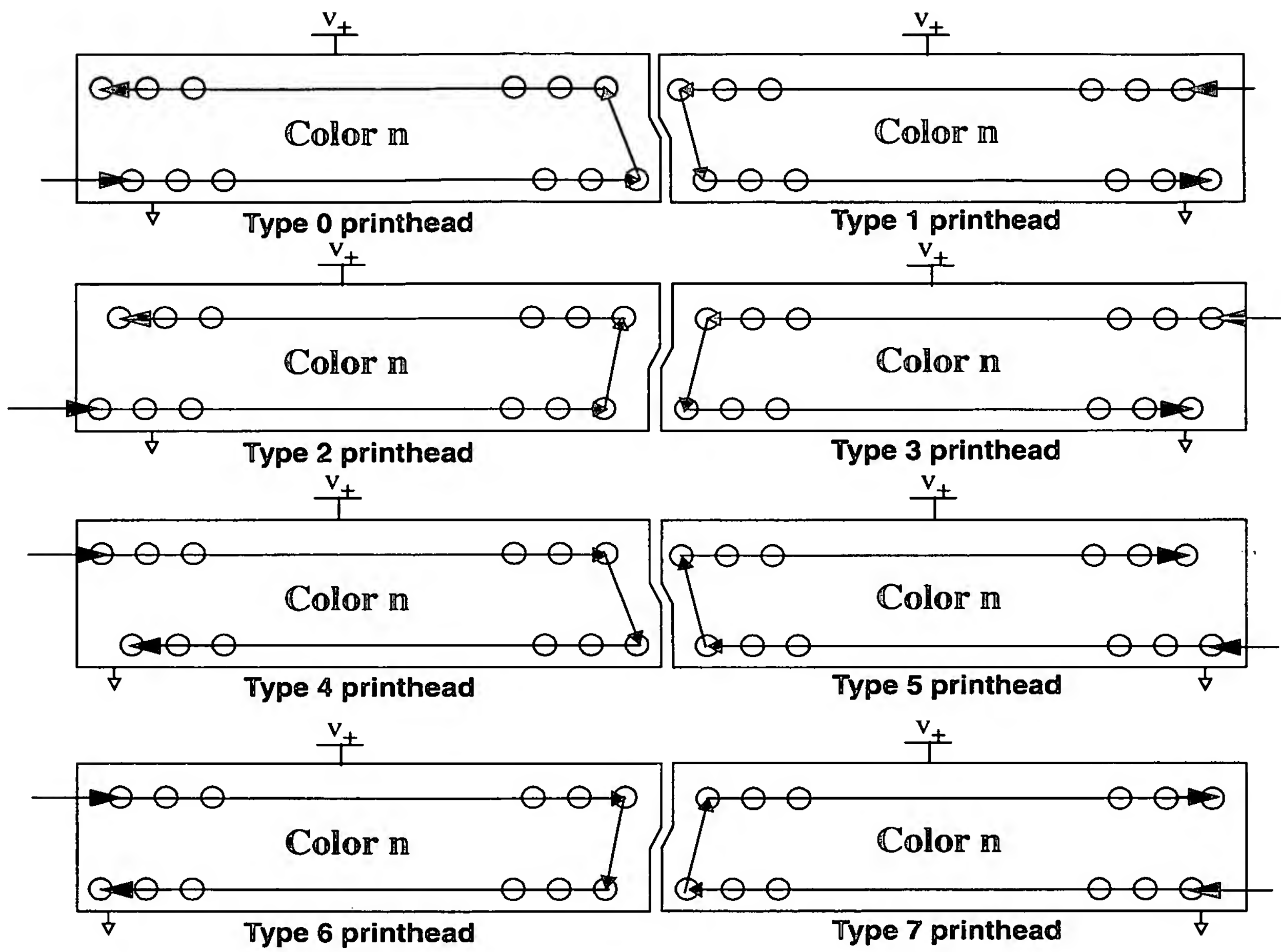


FIG. 295

255/331

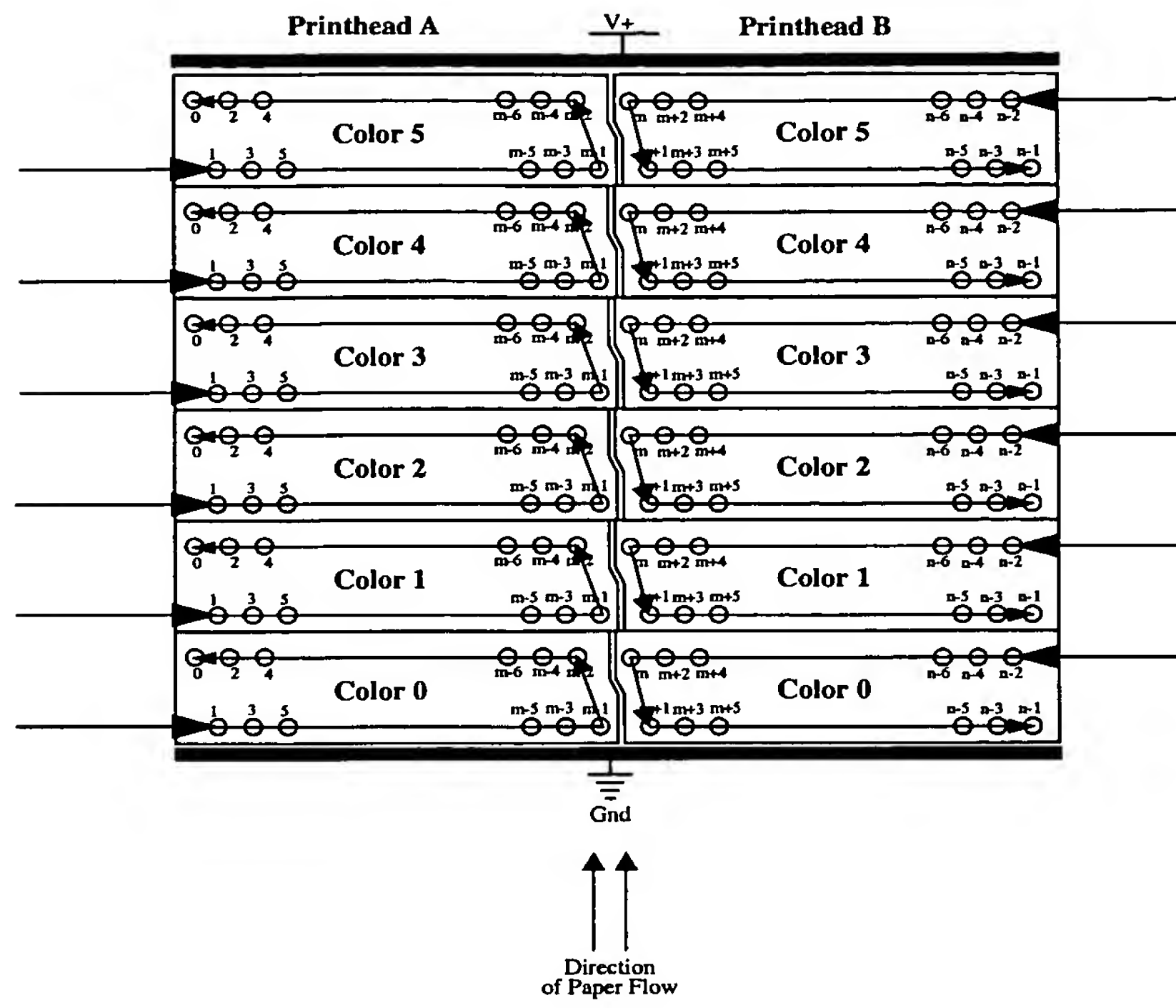


FIG. 296

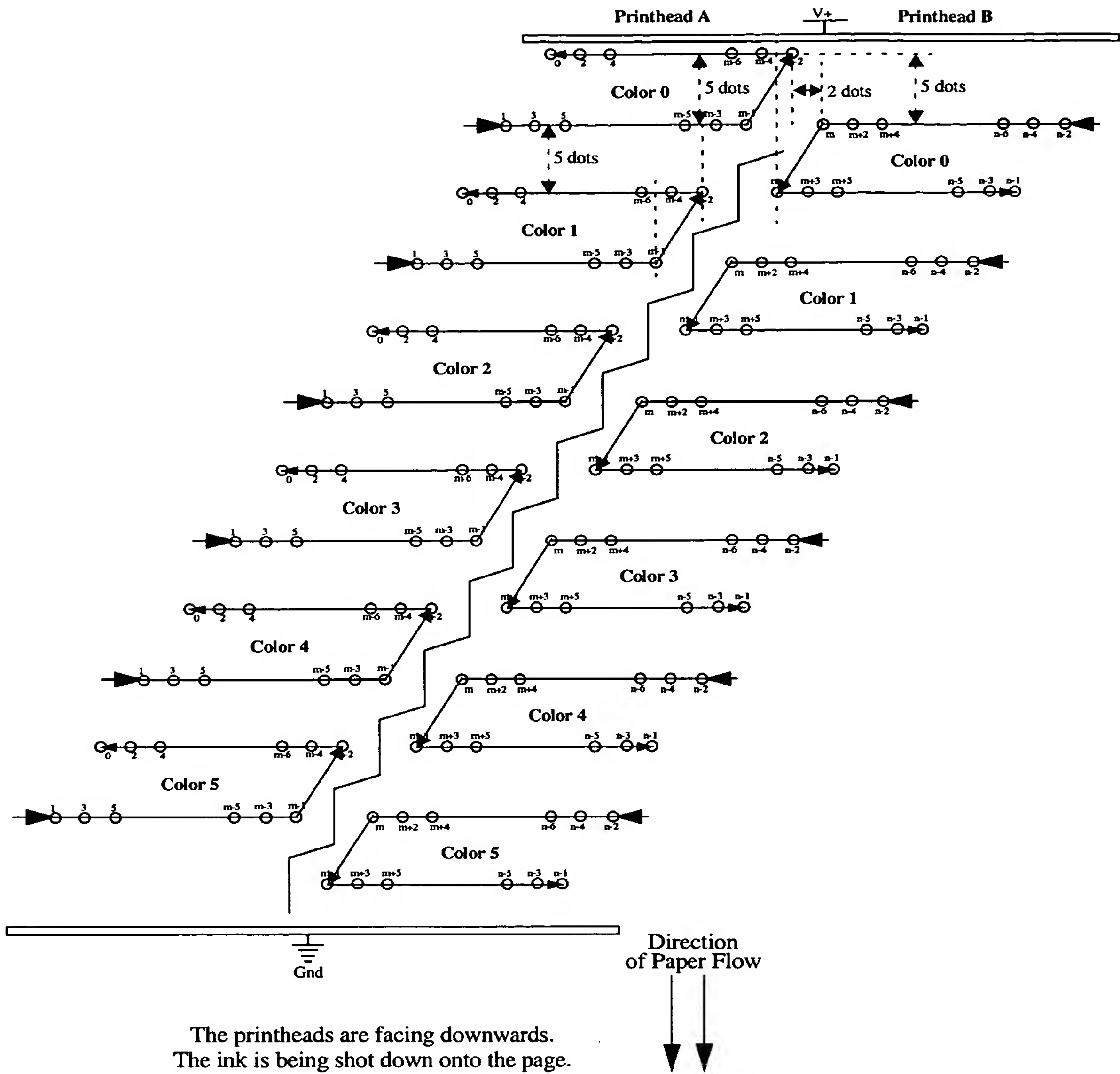


FIG. 297

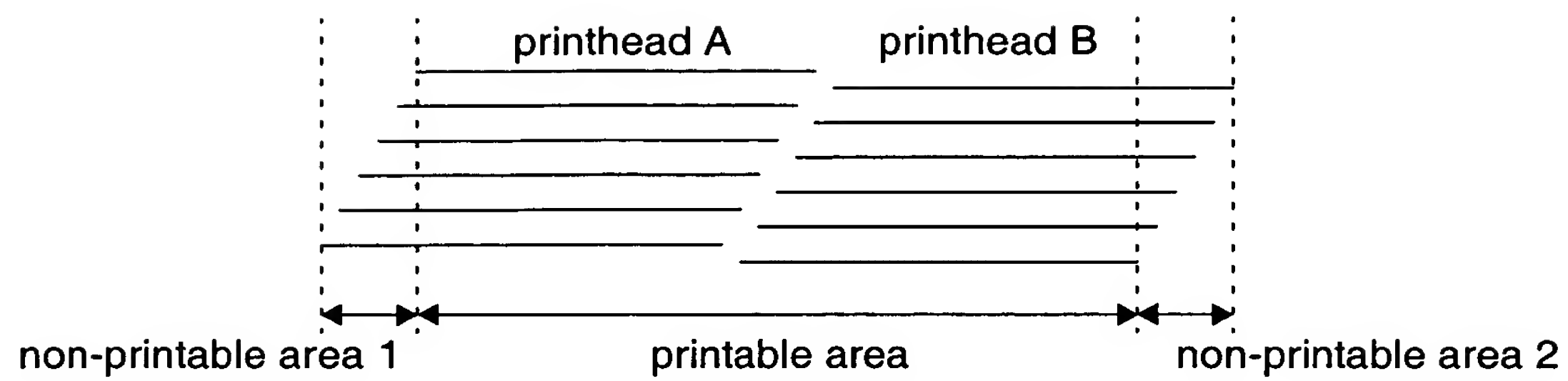


FIG. 298

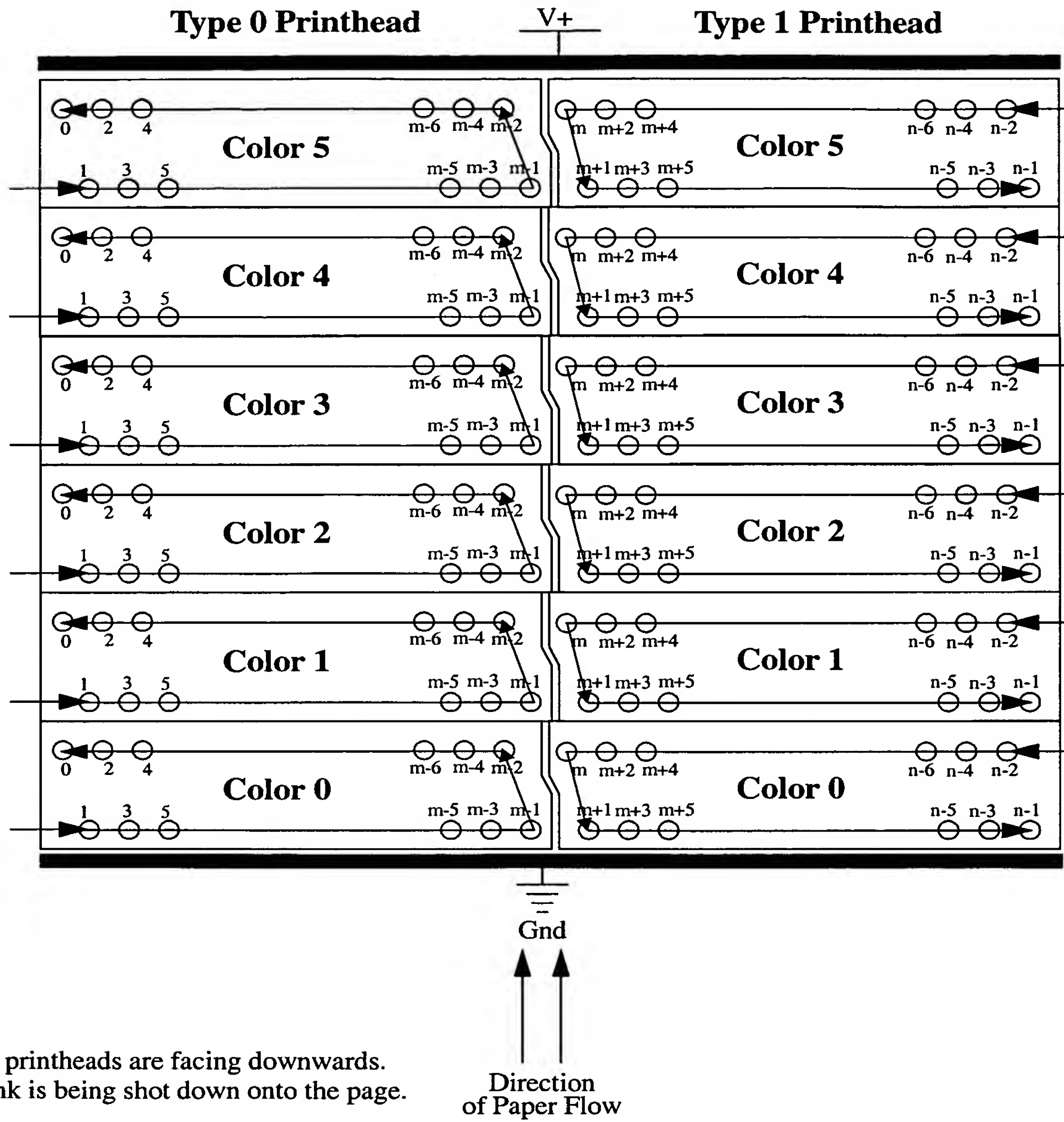


FIG. 299

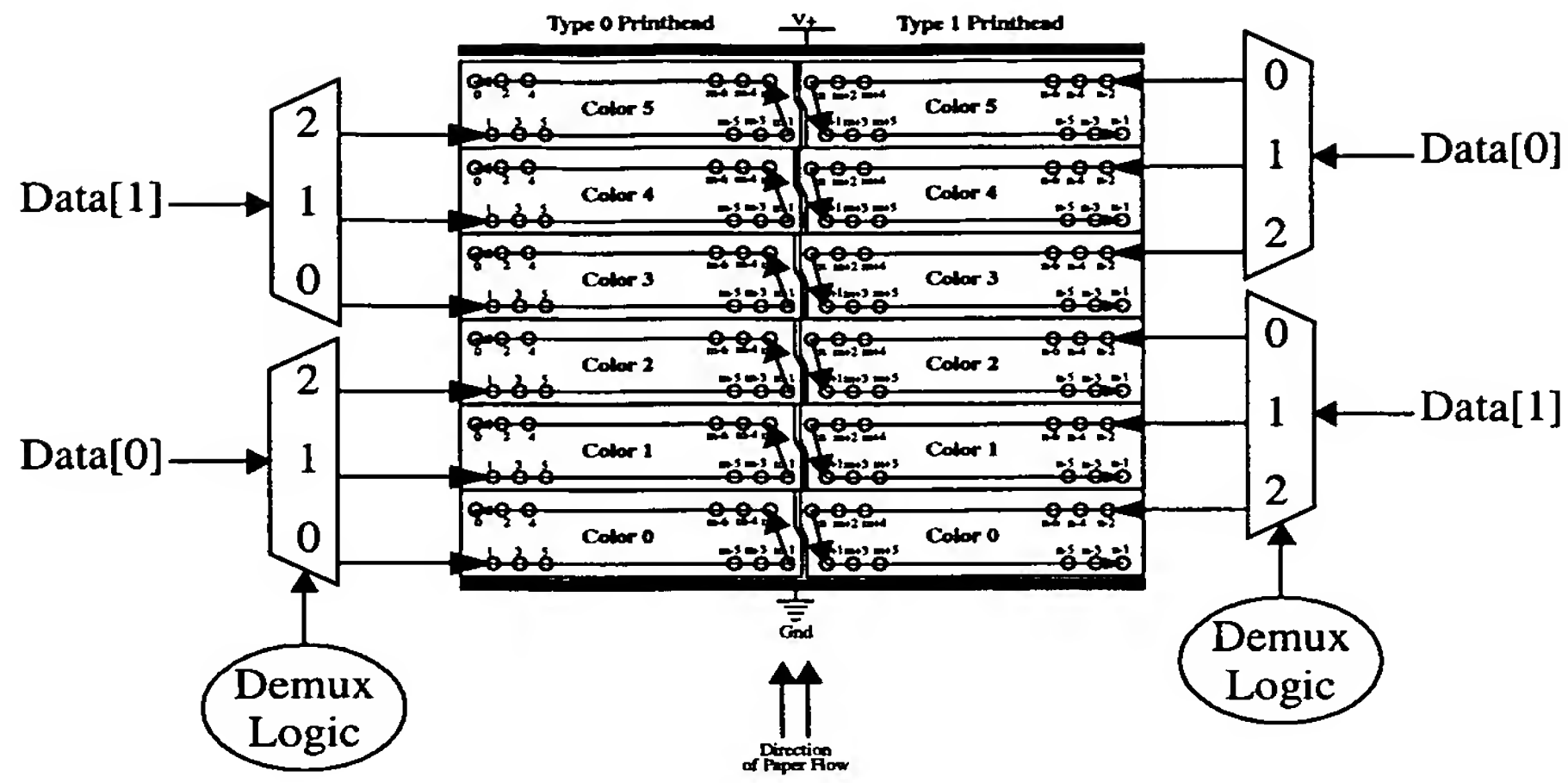


FIG. 300

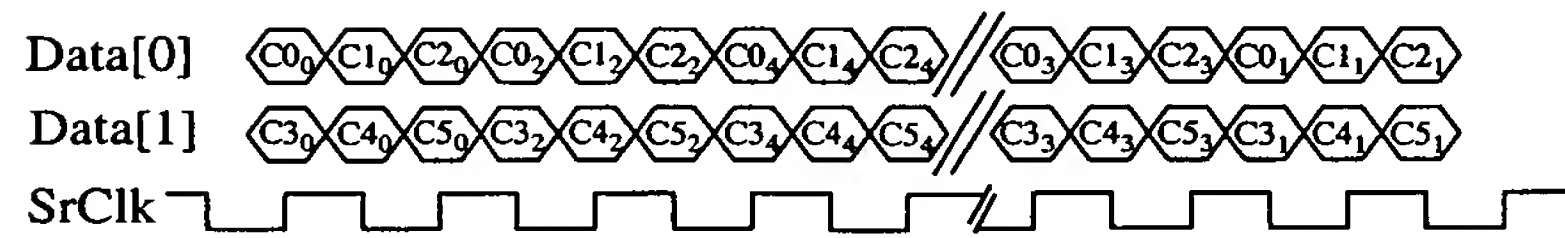


FIG. 301

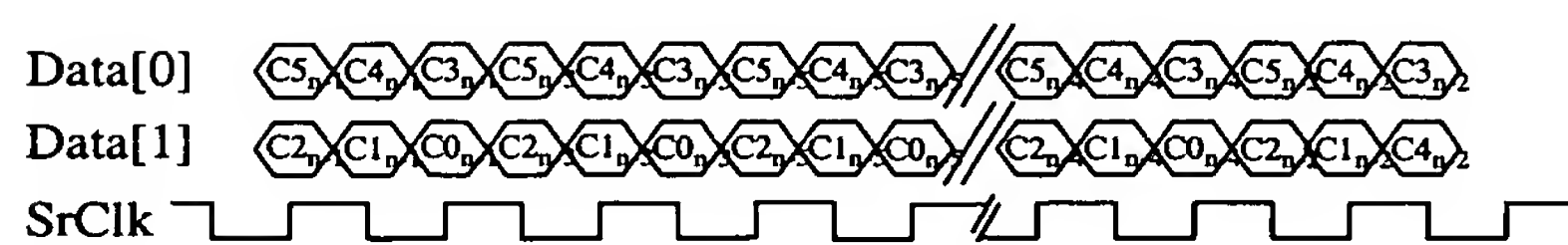


FIG. 302

260/331

The printheads are facing downwards.
The ink is being shot down onto the page.

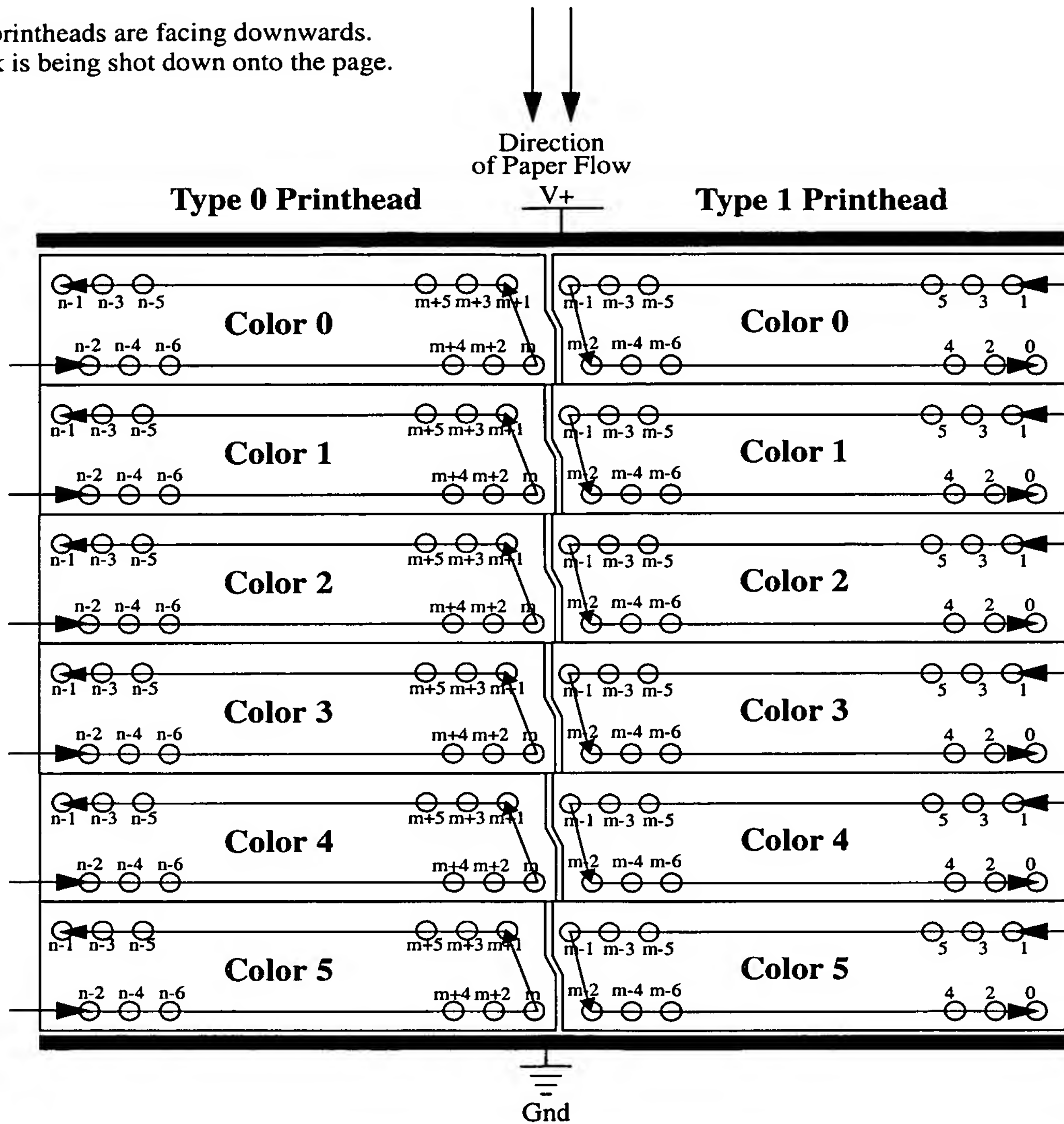


FIG. 303

261/331

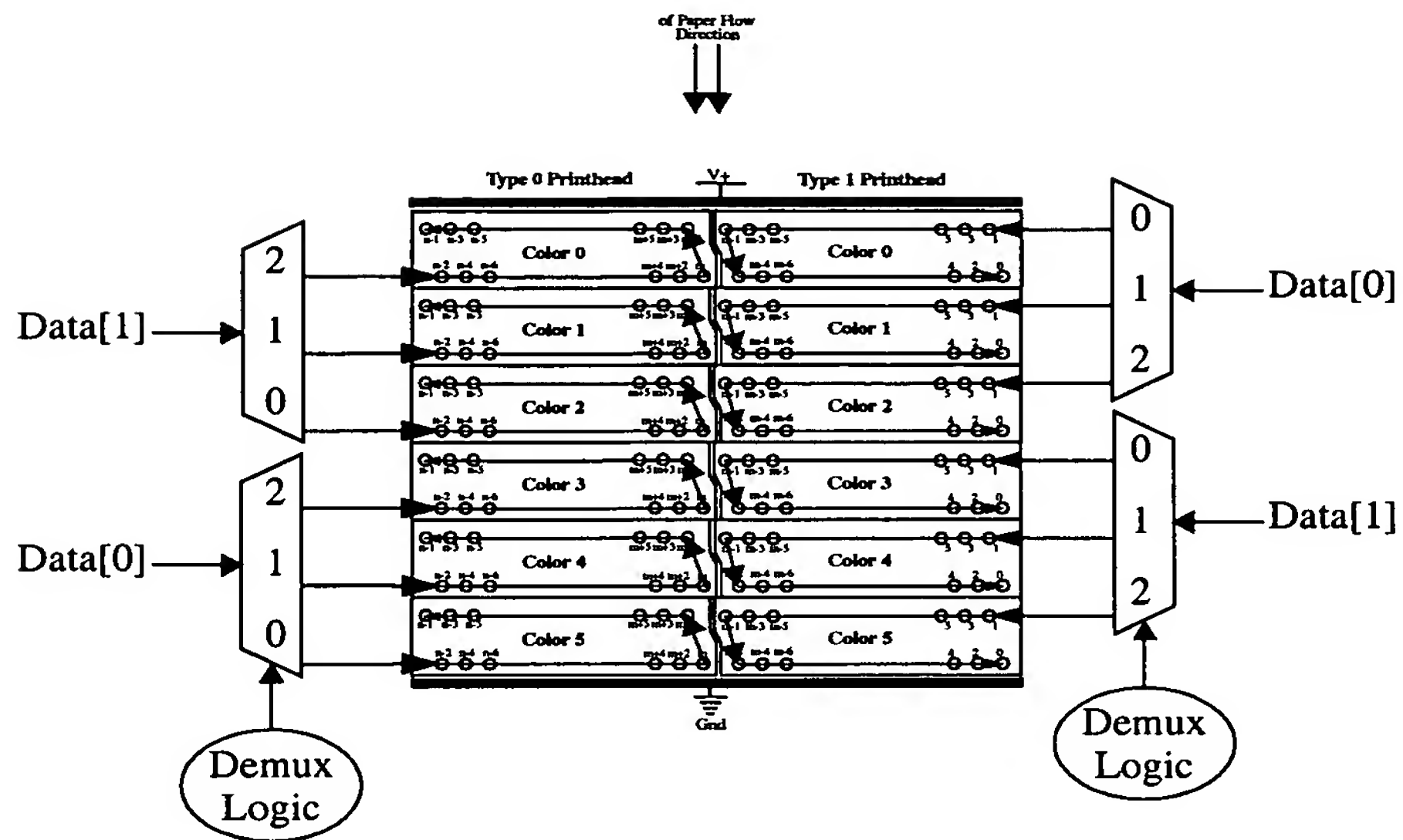


FIG. 304

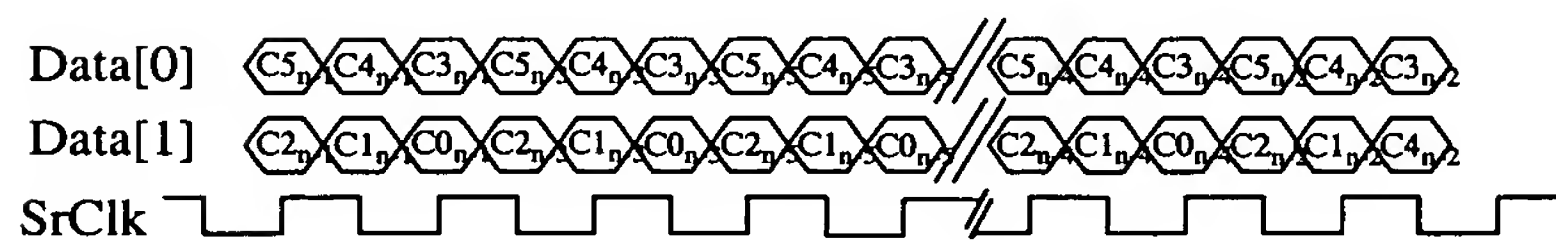


FIG. 305

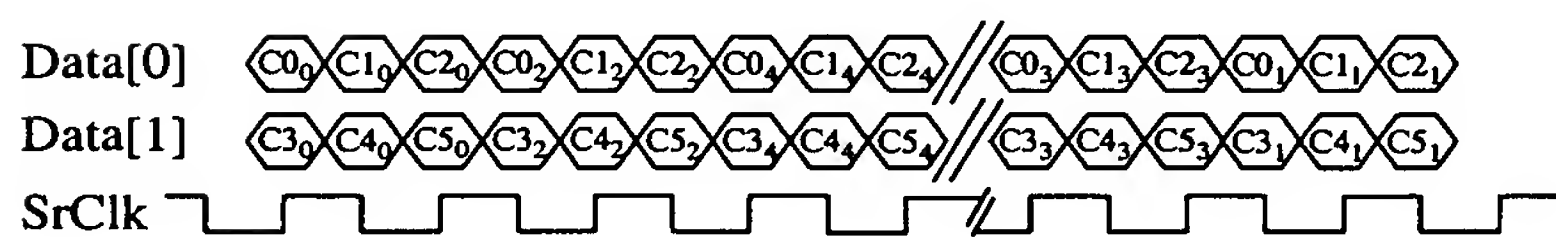


FIG. 306

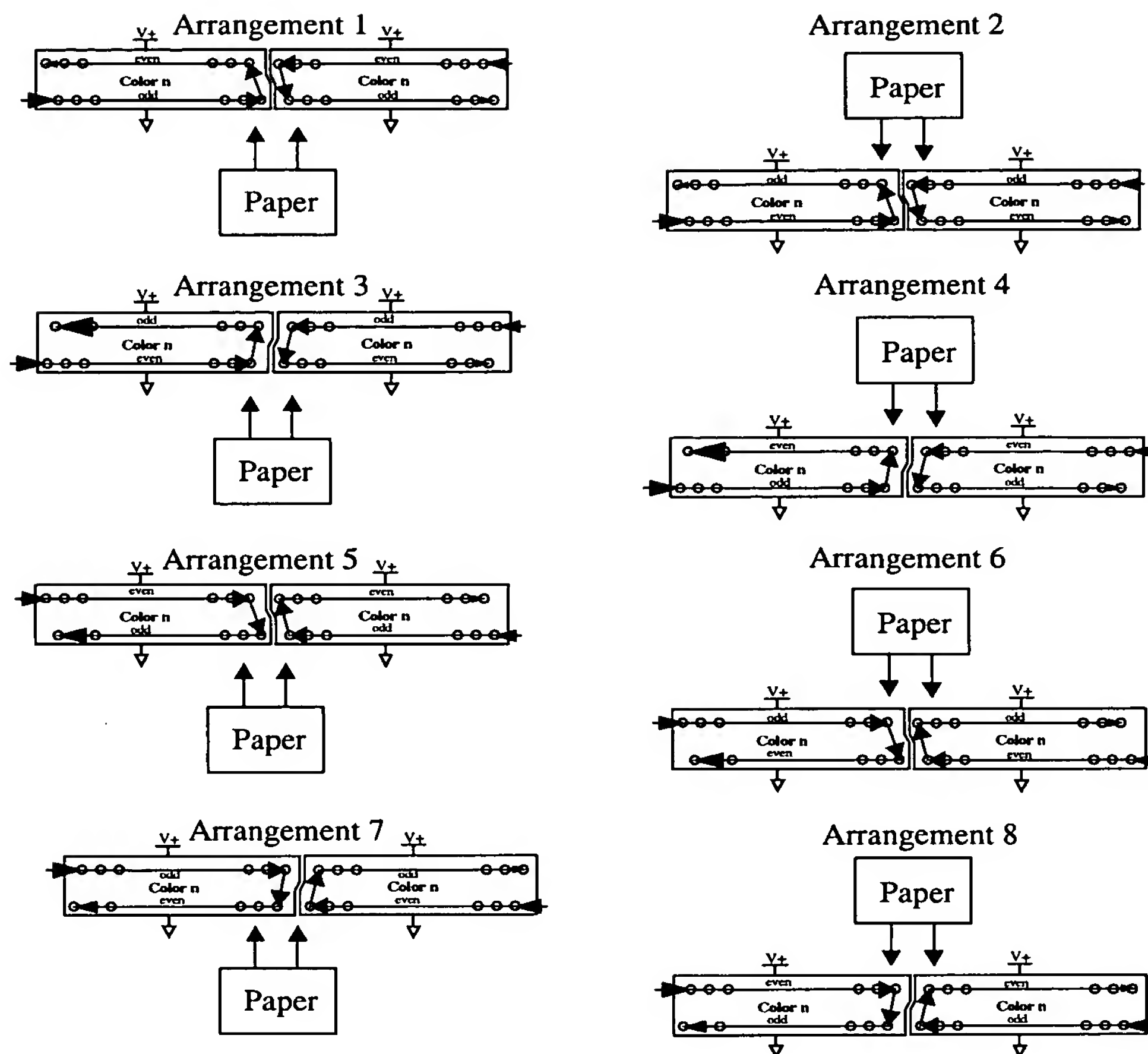


FIG. 307

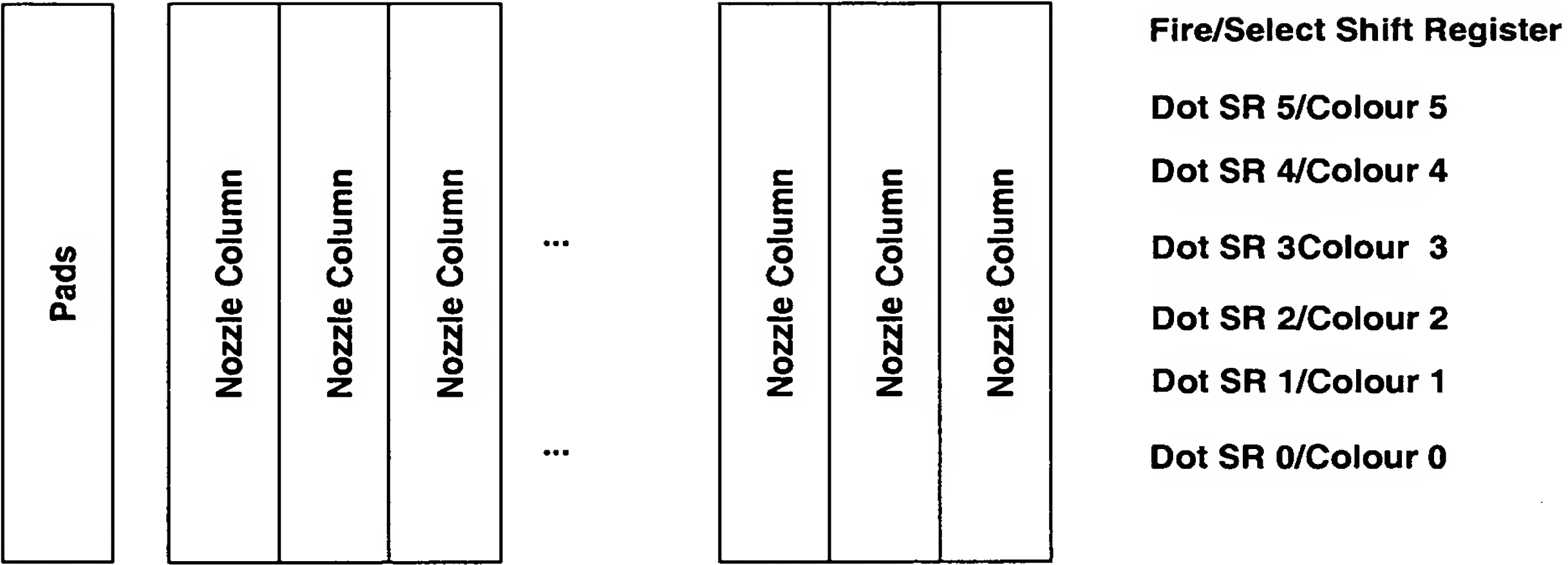


FIG. 308

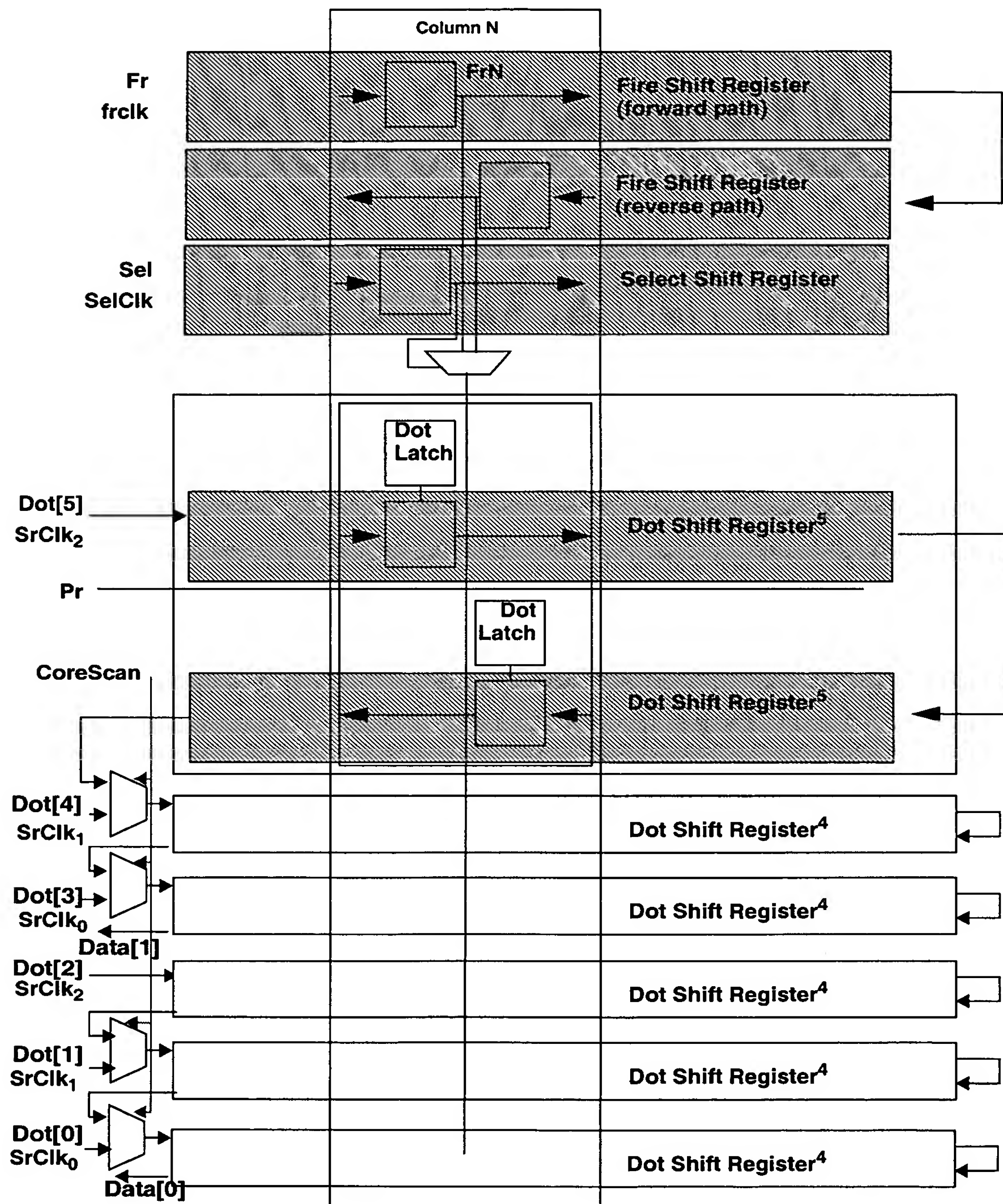


FIG. 309

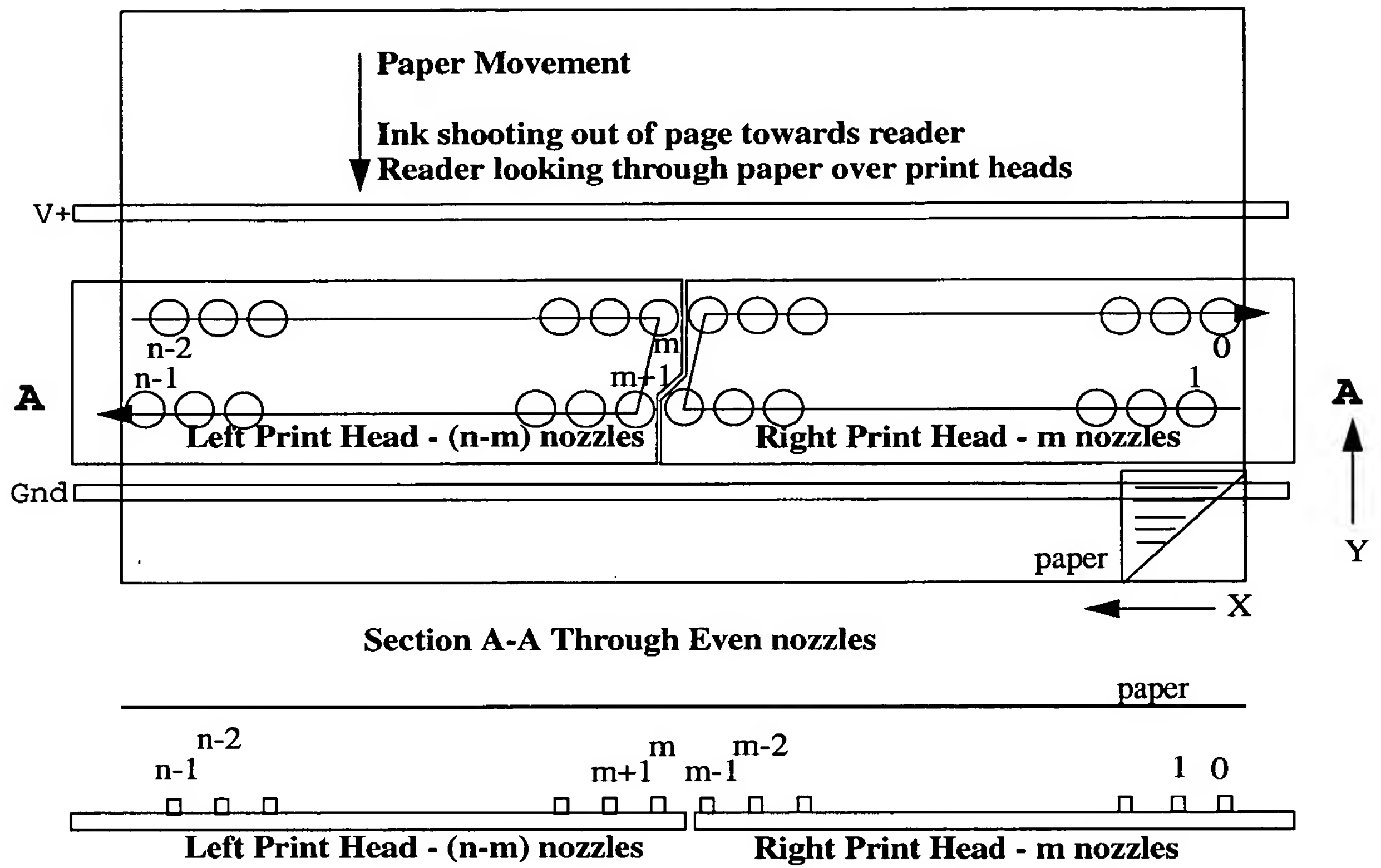


FIG. 310

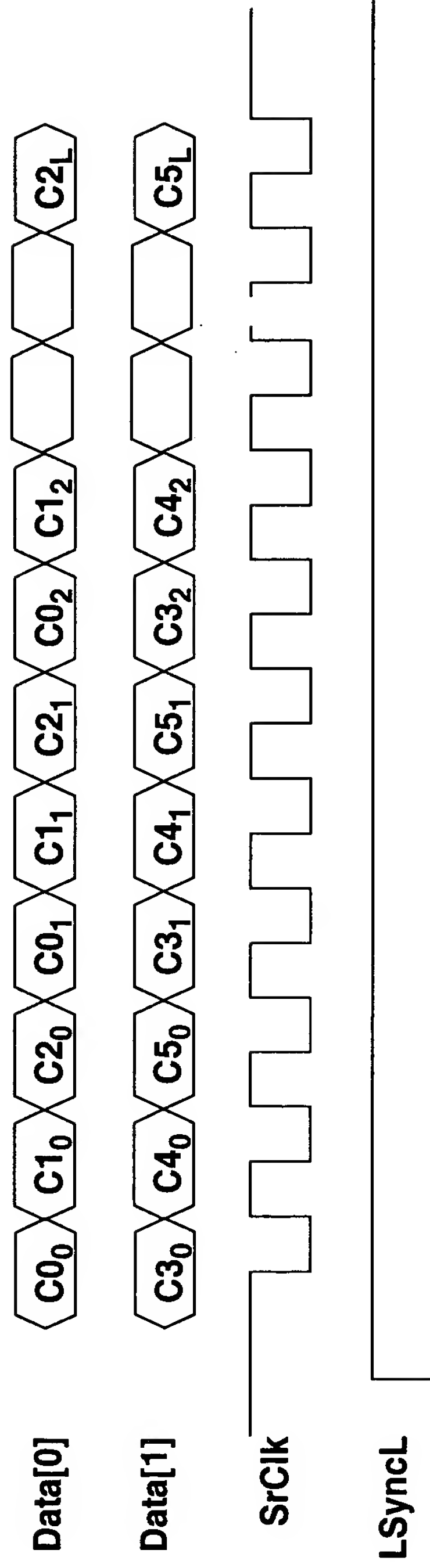
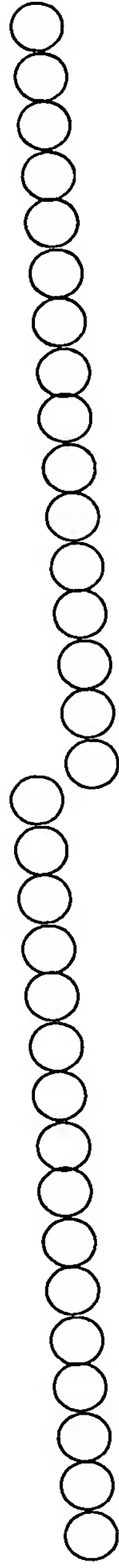
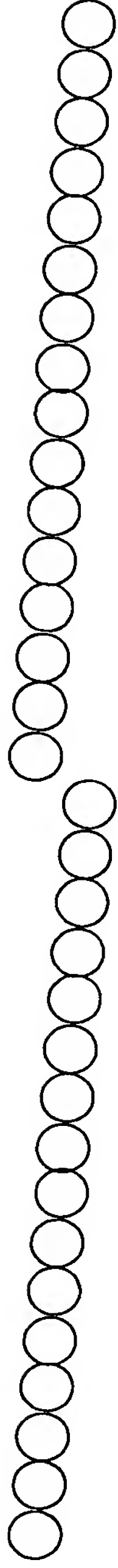


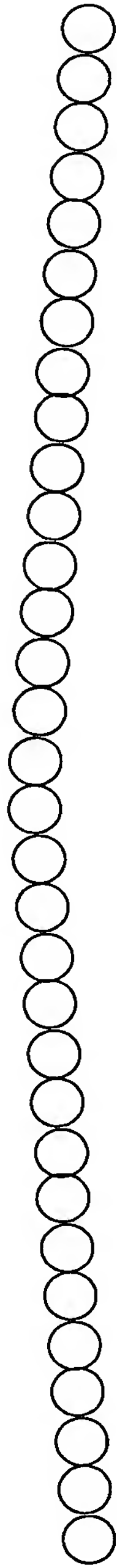
FIG. 311



a) Printing every n^{th} dot with all zero's in the fire select shift register



b) Printing every n^{th} dot with all one's in the fire select shift register



c) Printing every n^{th} dot with n zero's then n one's in the fire select shift registers

FIG. 312

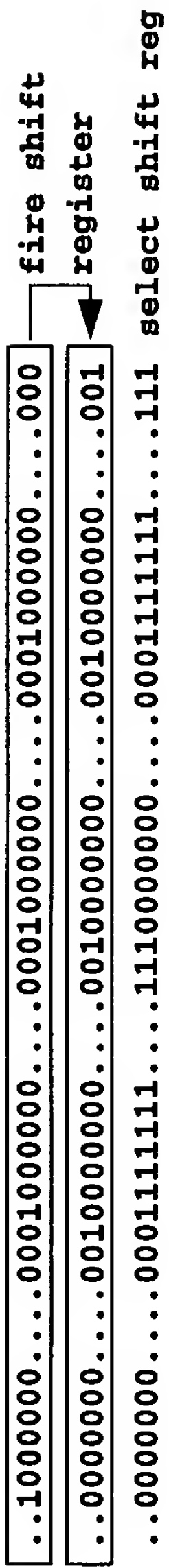


FIG. 313

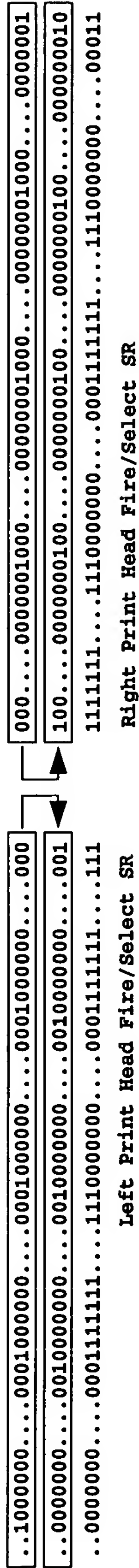


FIG. 314

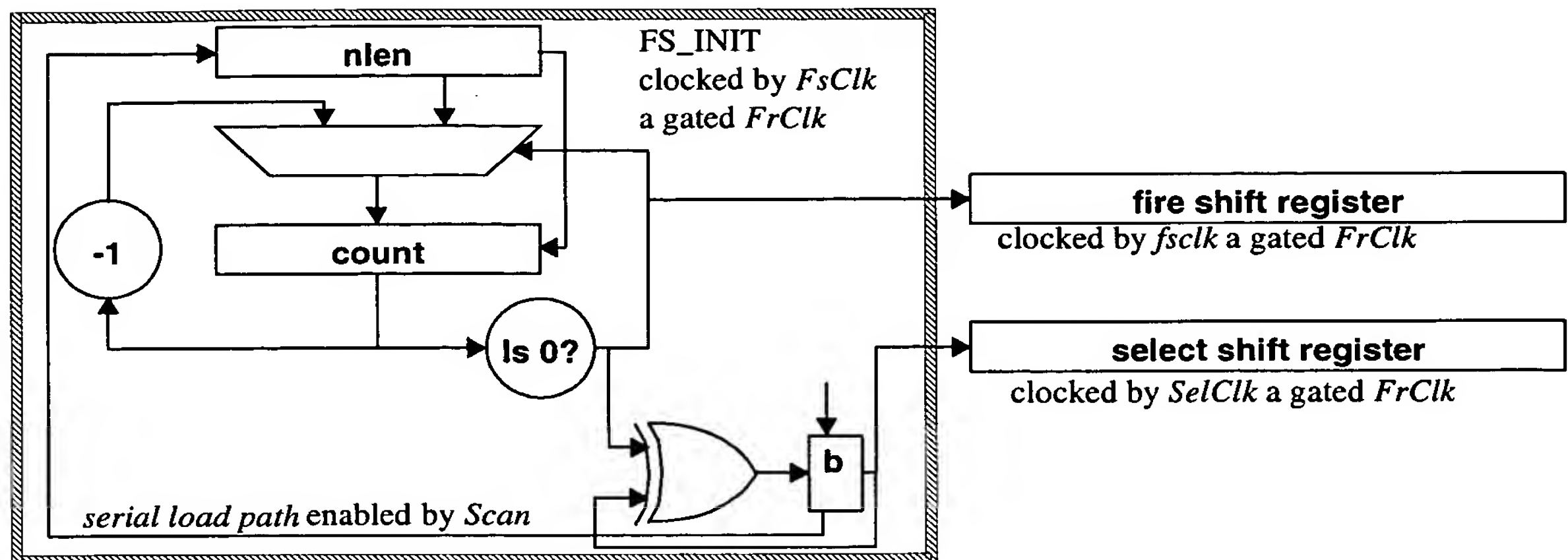


FIG. 315

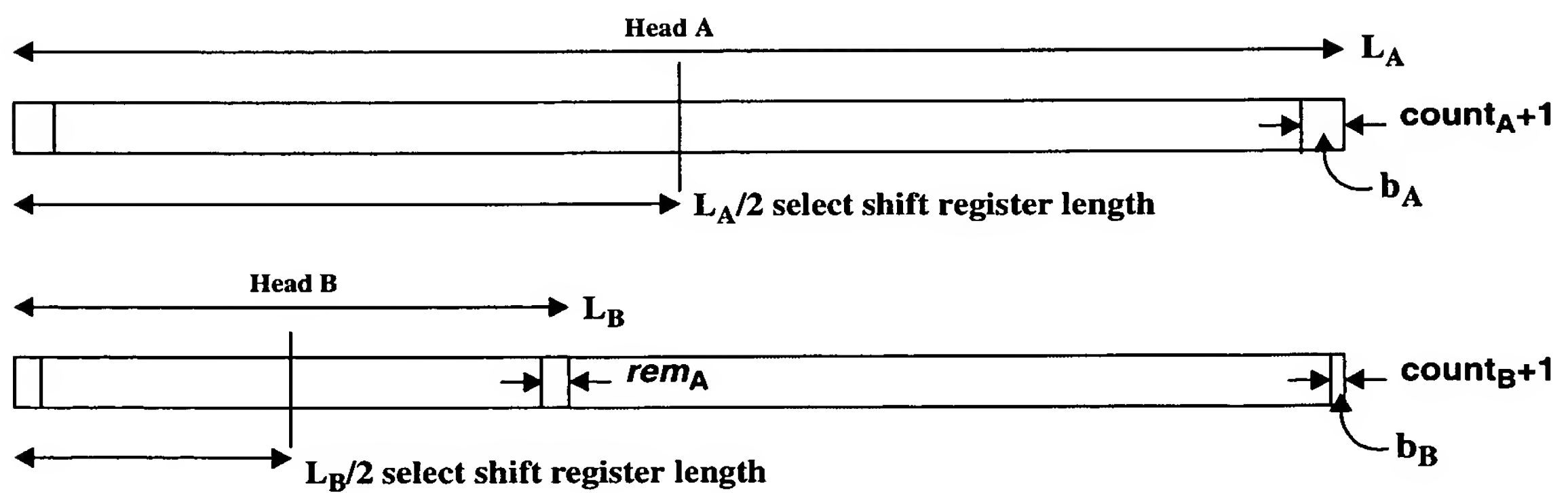


FIG. 316

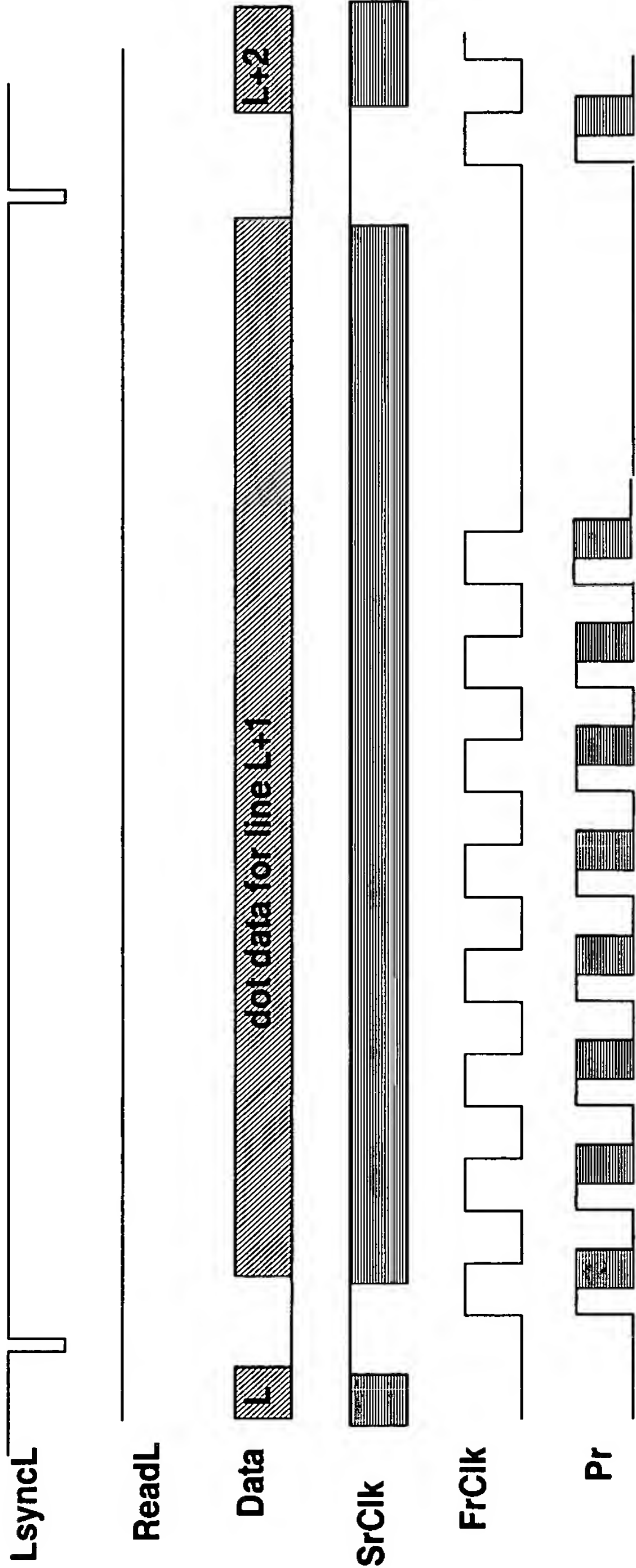


FIG. 317

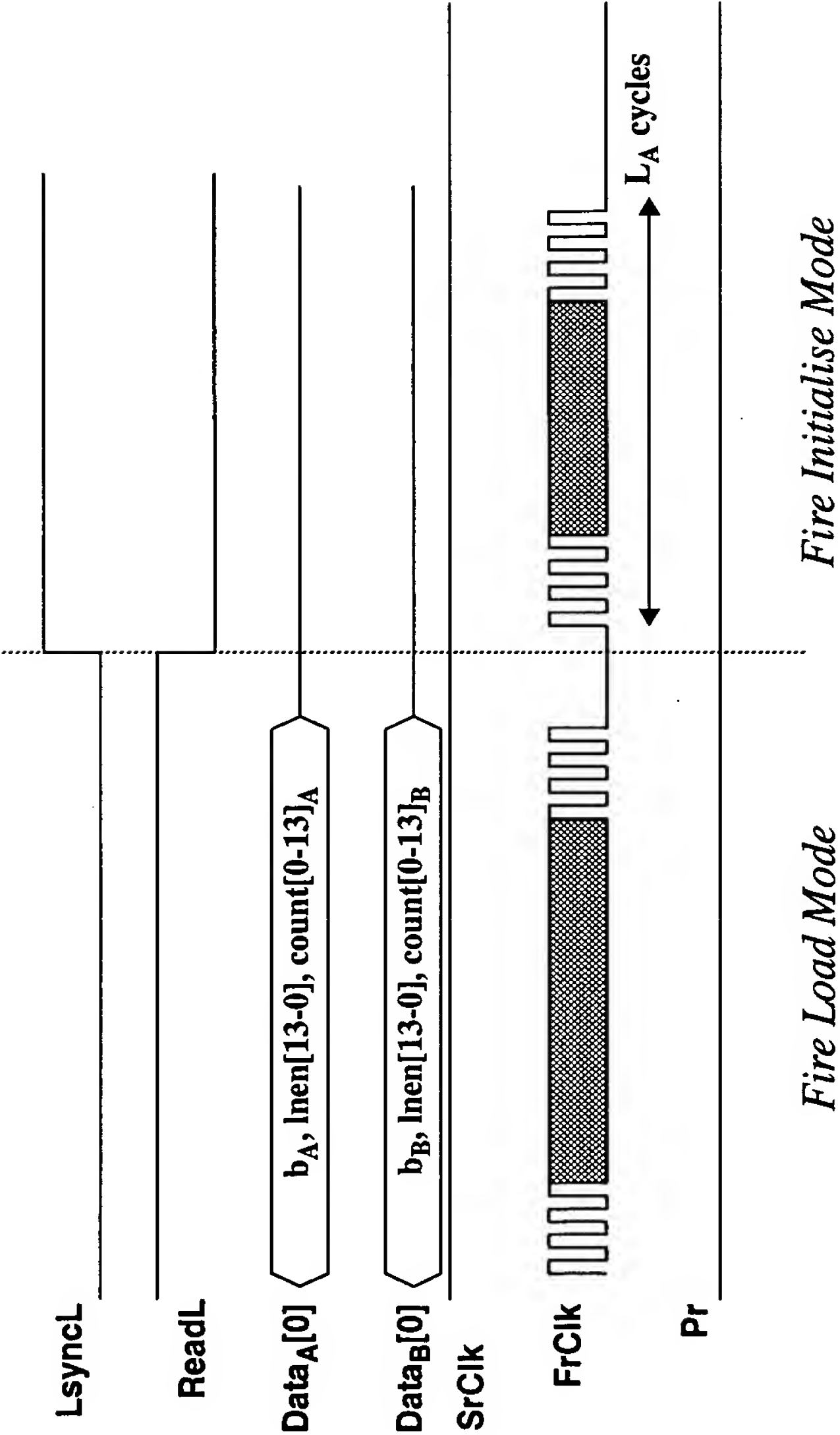


FIG. 318

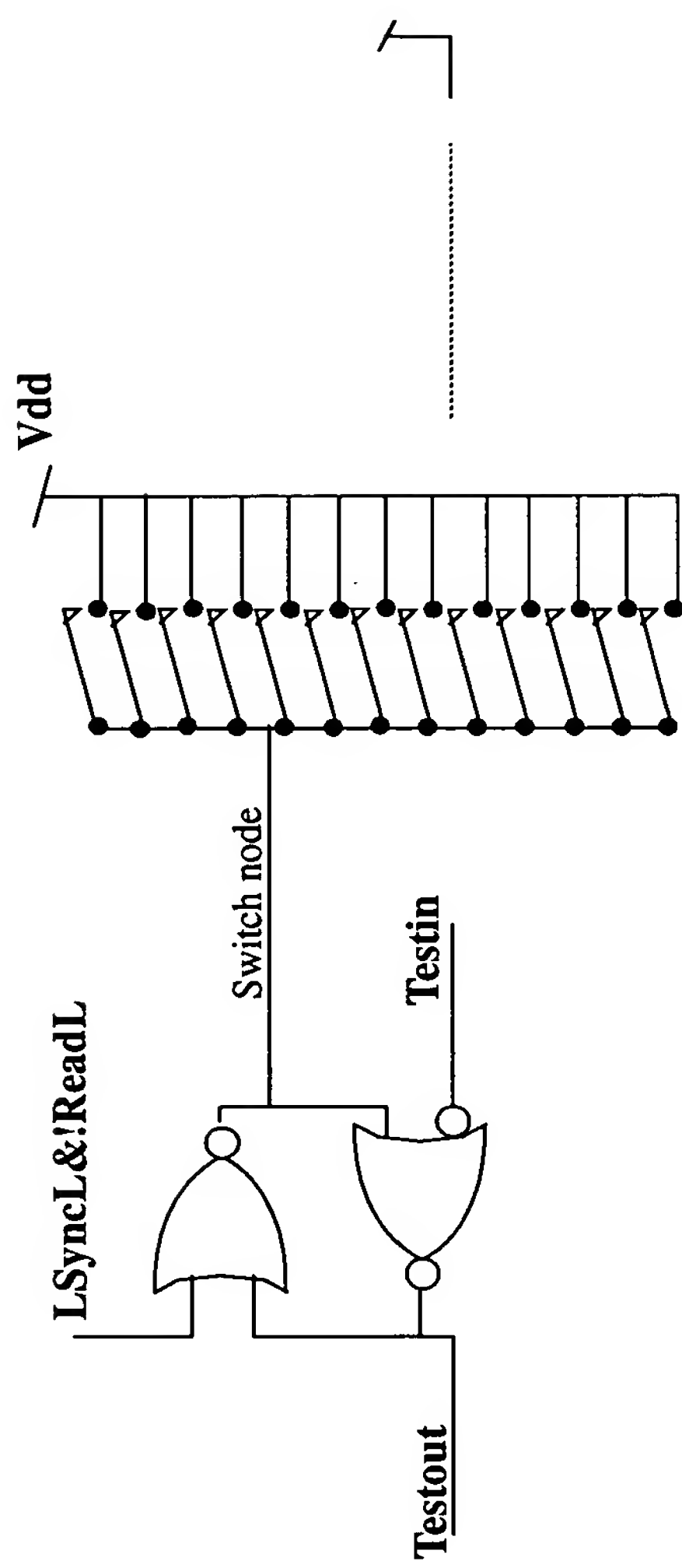


FIG. 319

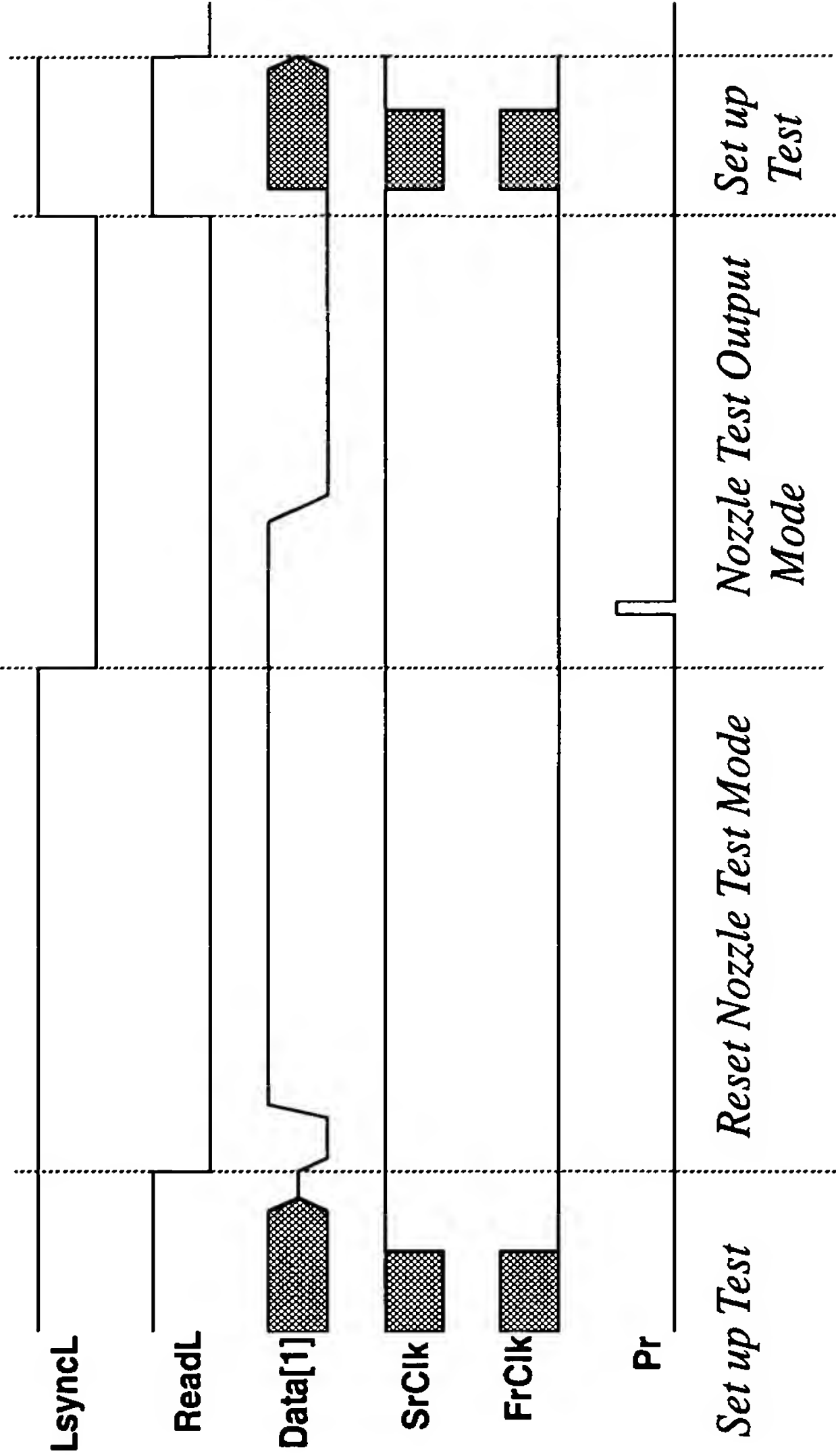


FIG. 320

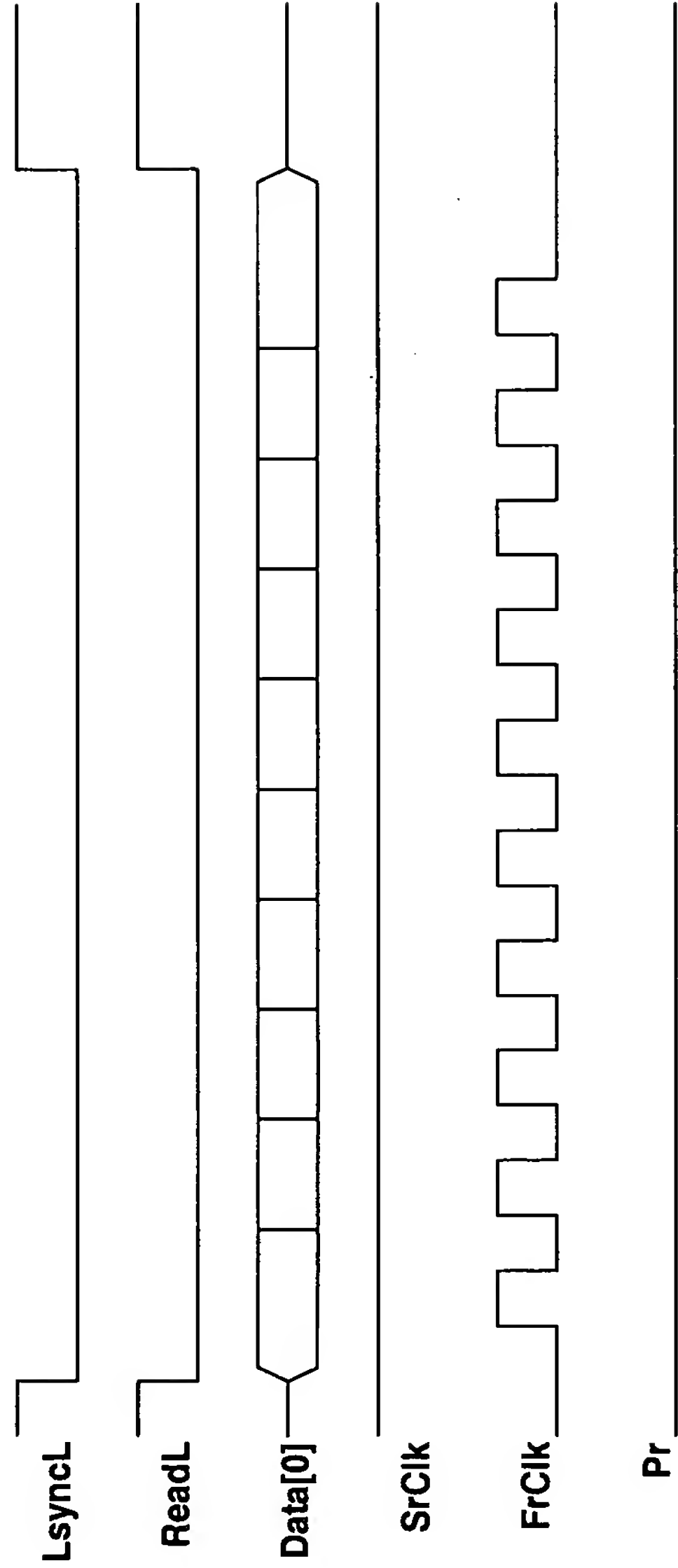


FIG. 321

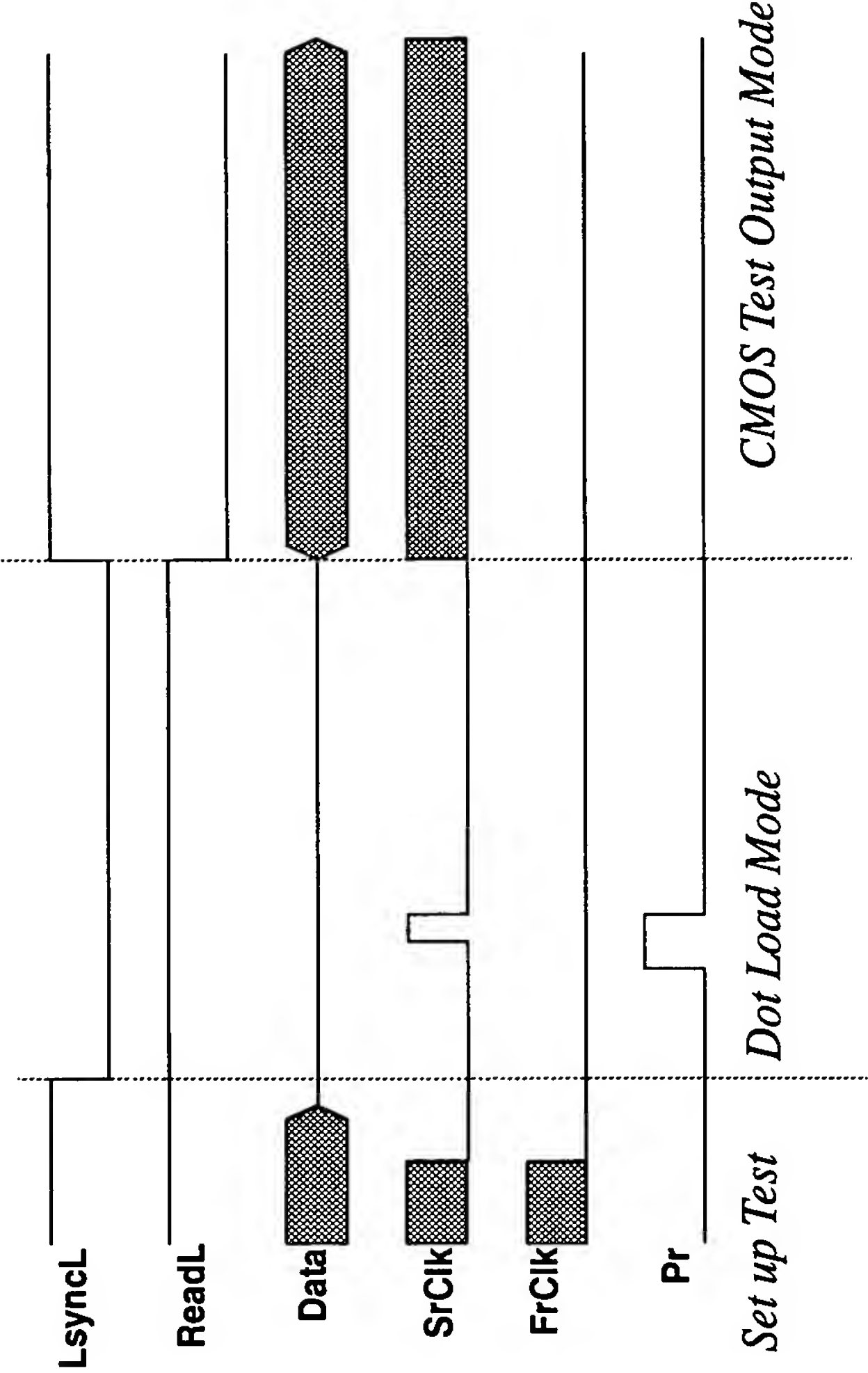


FIG. 322

276/331

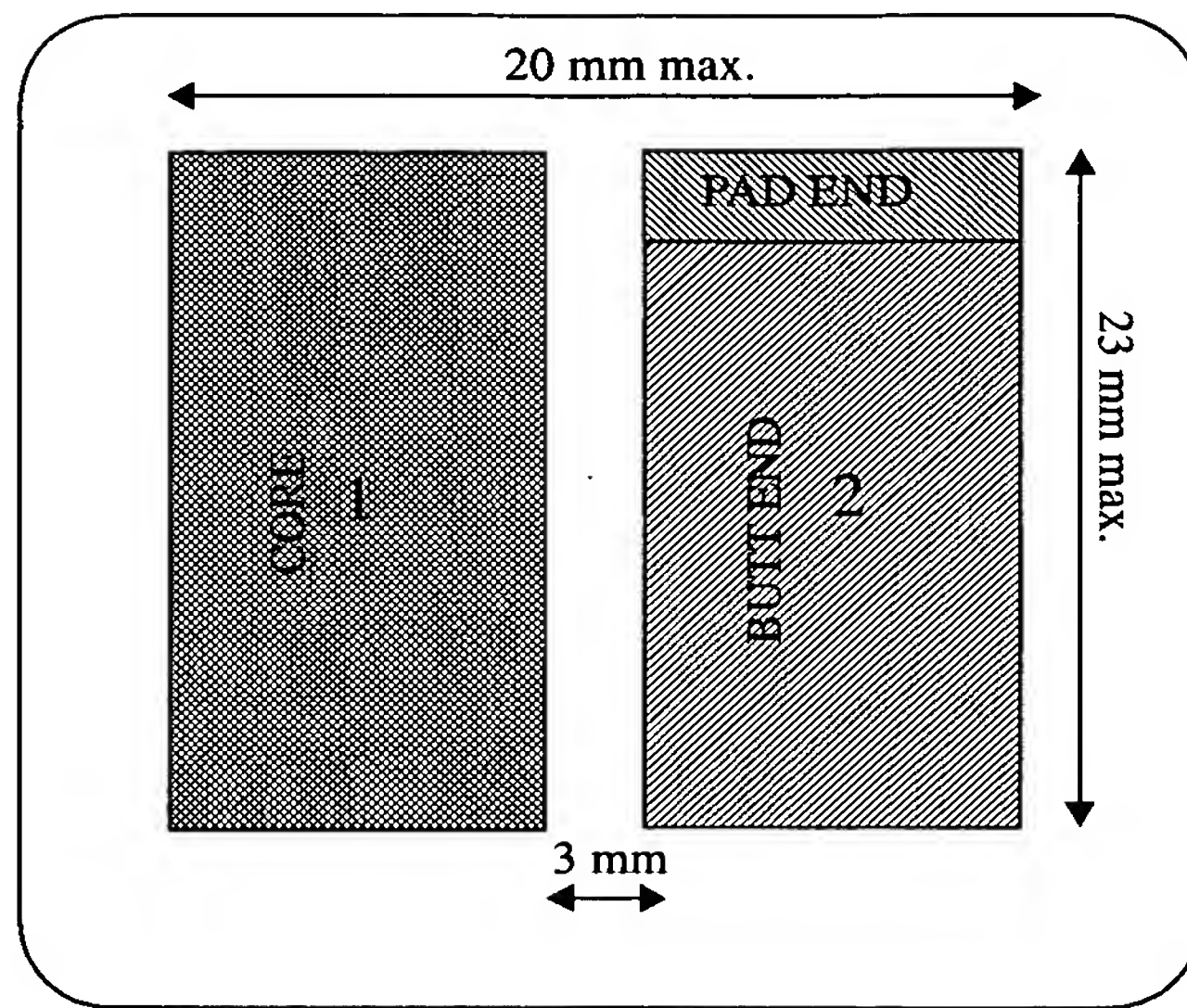


FIG. 323

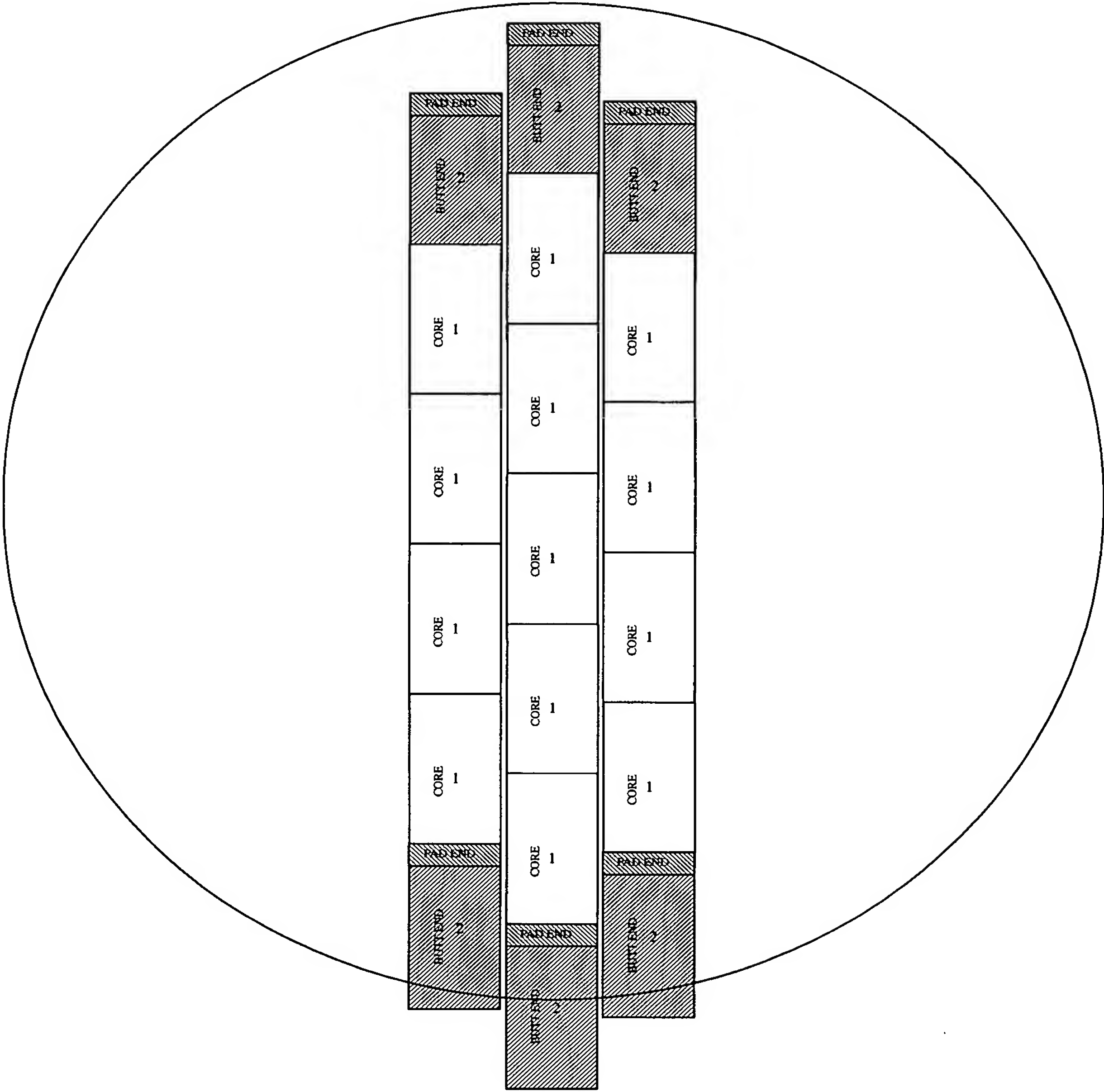


FIG. 324

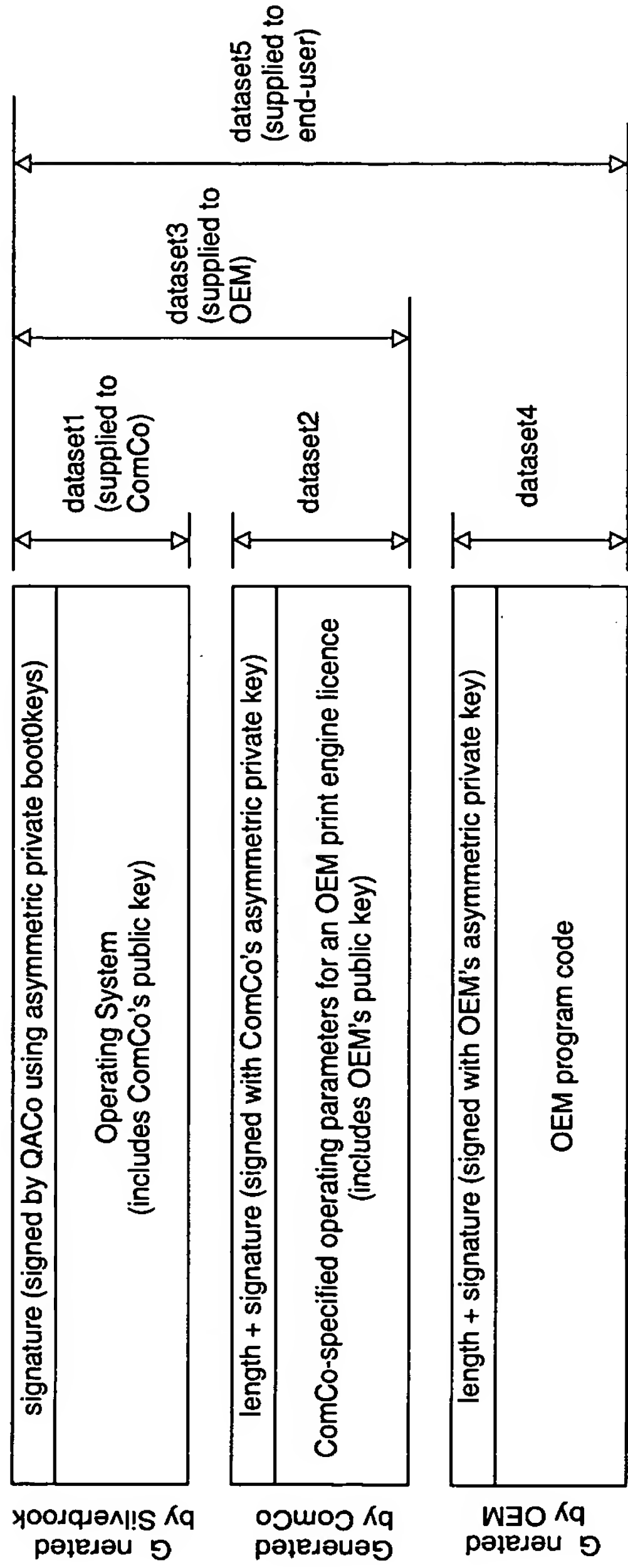


FIG. 325

278/331

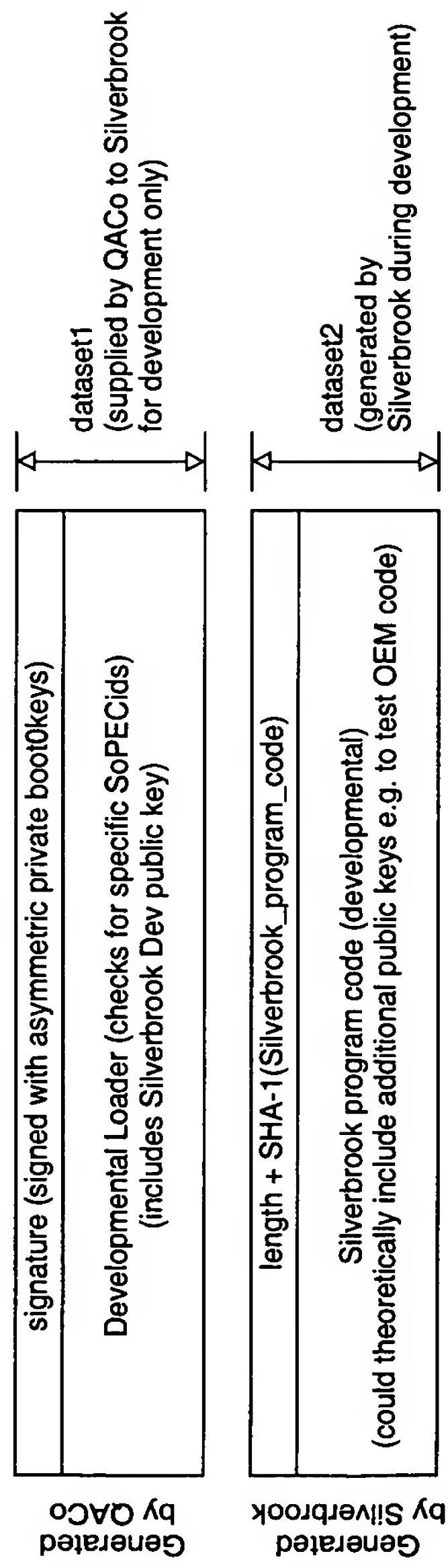


FIG. 327

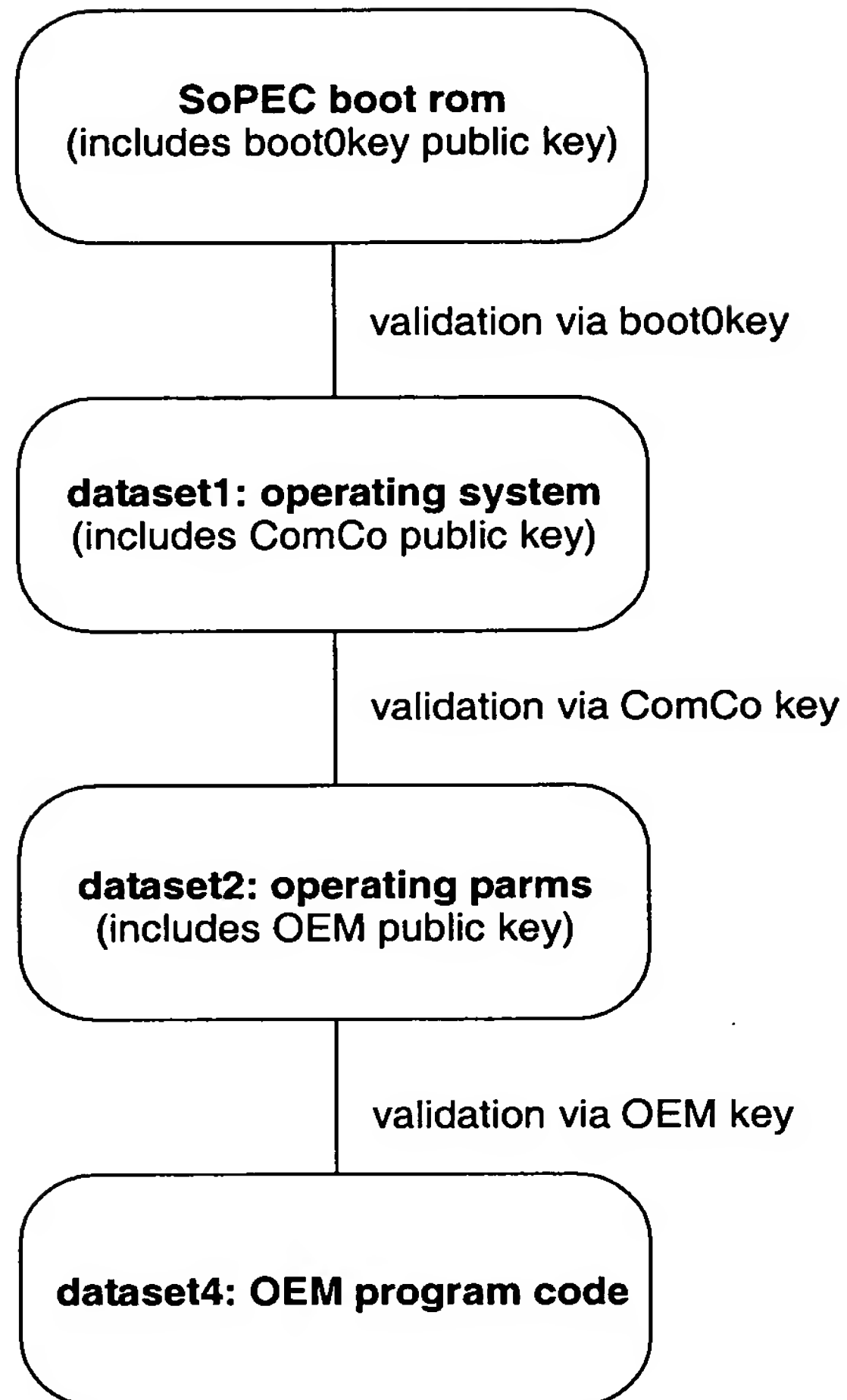


FIG. 326

280/331

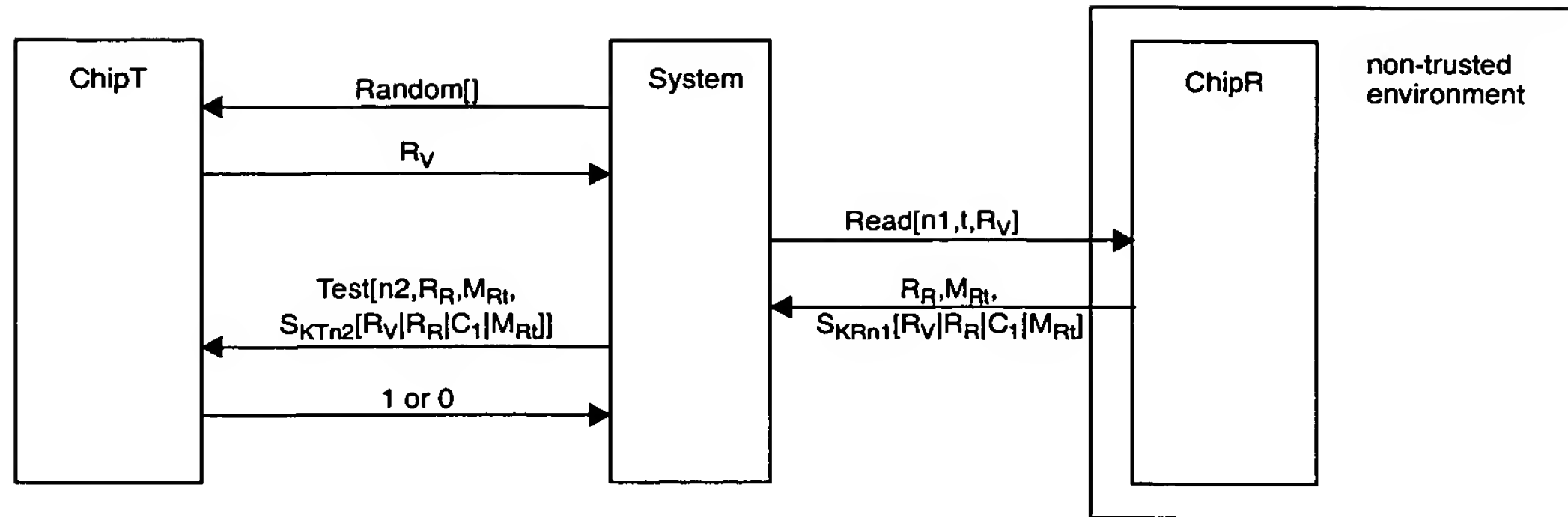


FIG. 328

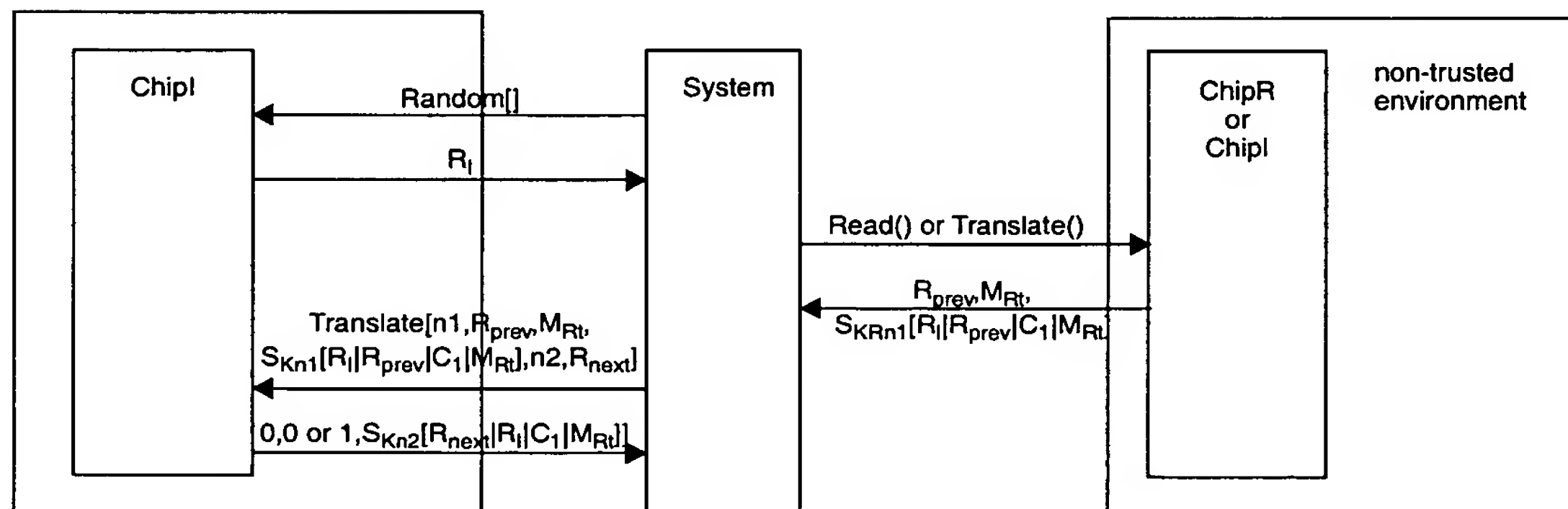


FIG. 329

281/331

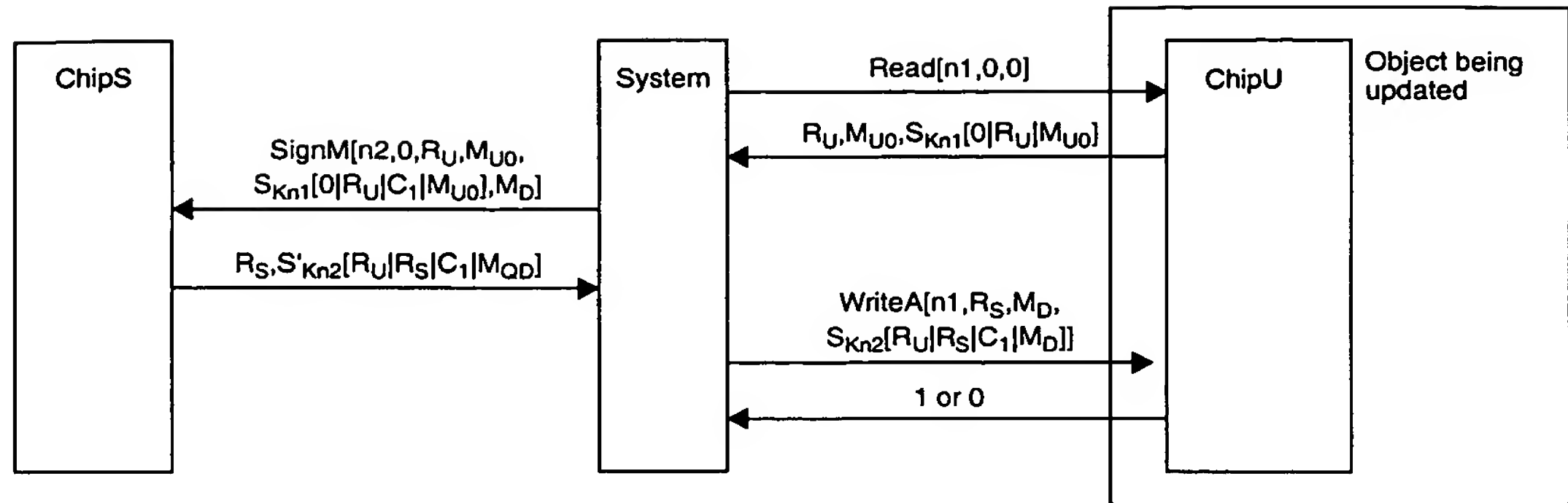


FIG. 330

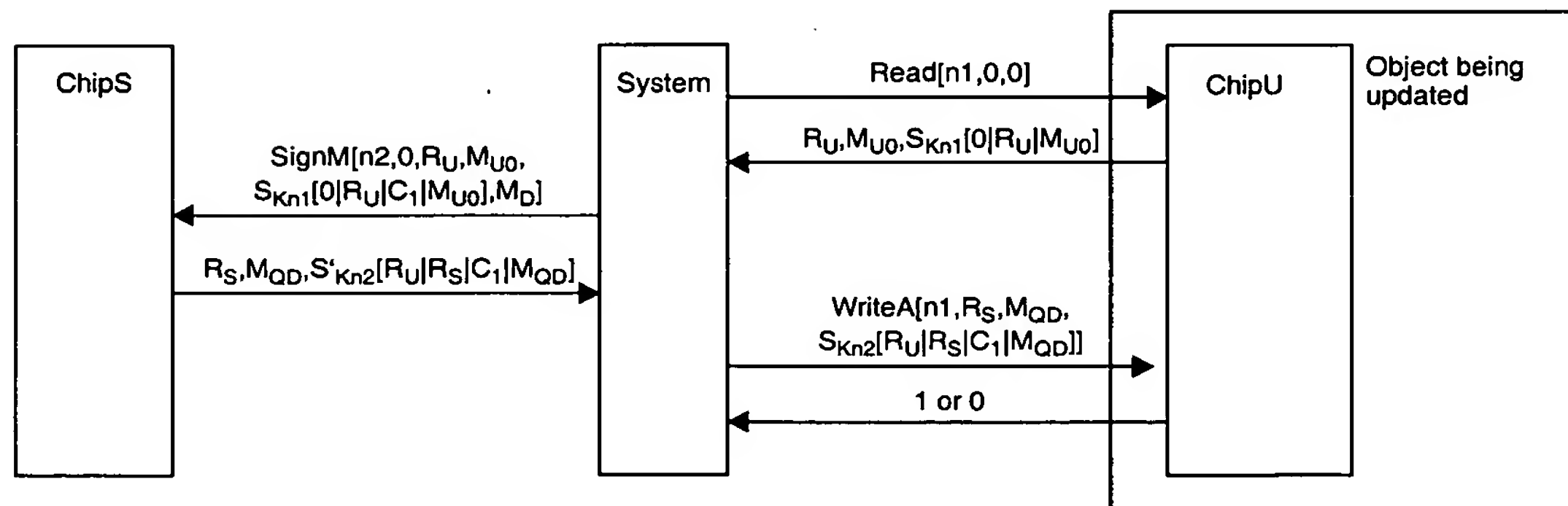


FIG. 331

282/331

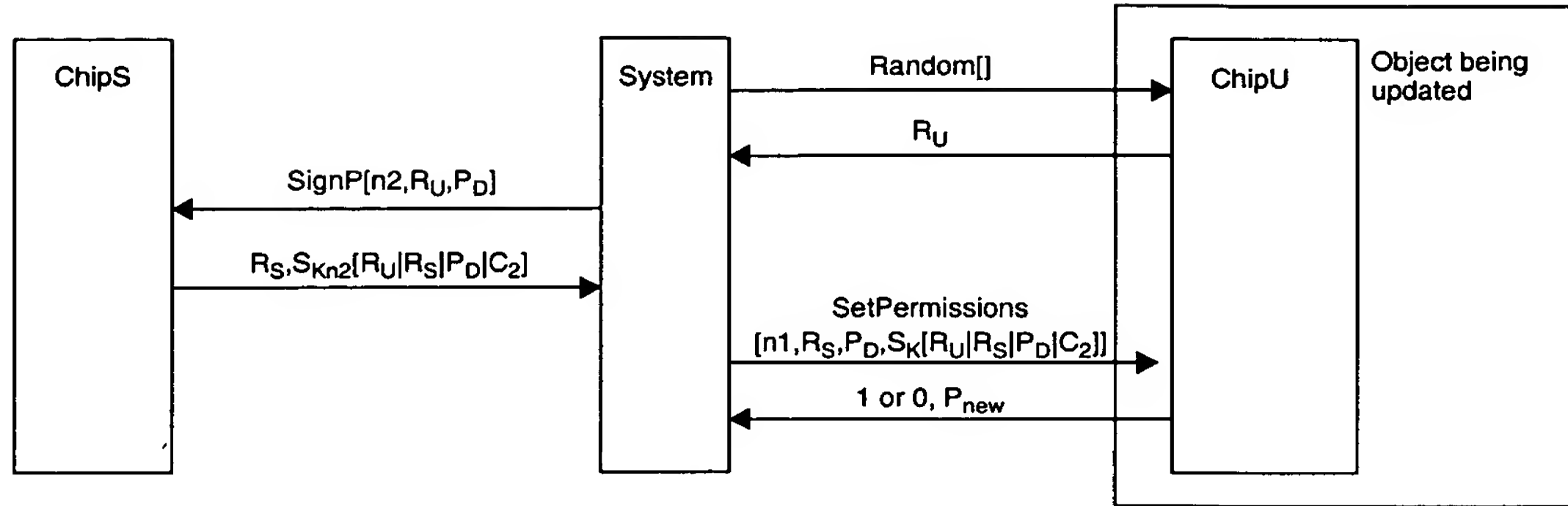


FIG. 332

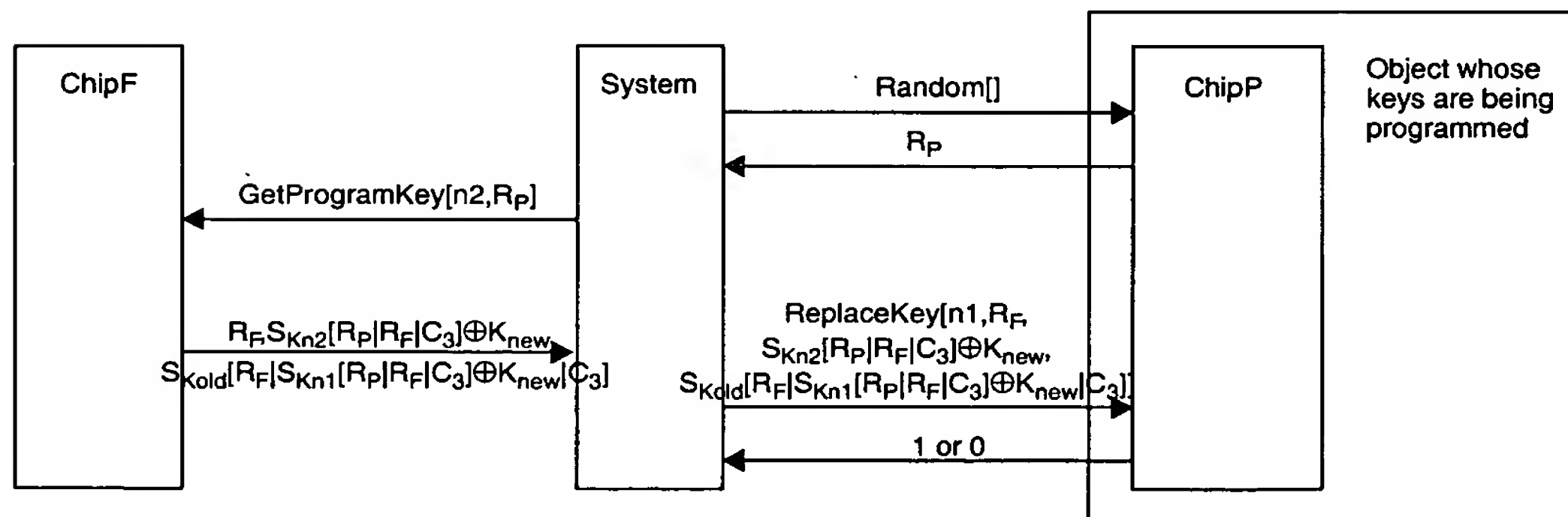


FIG. 333

283/331

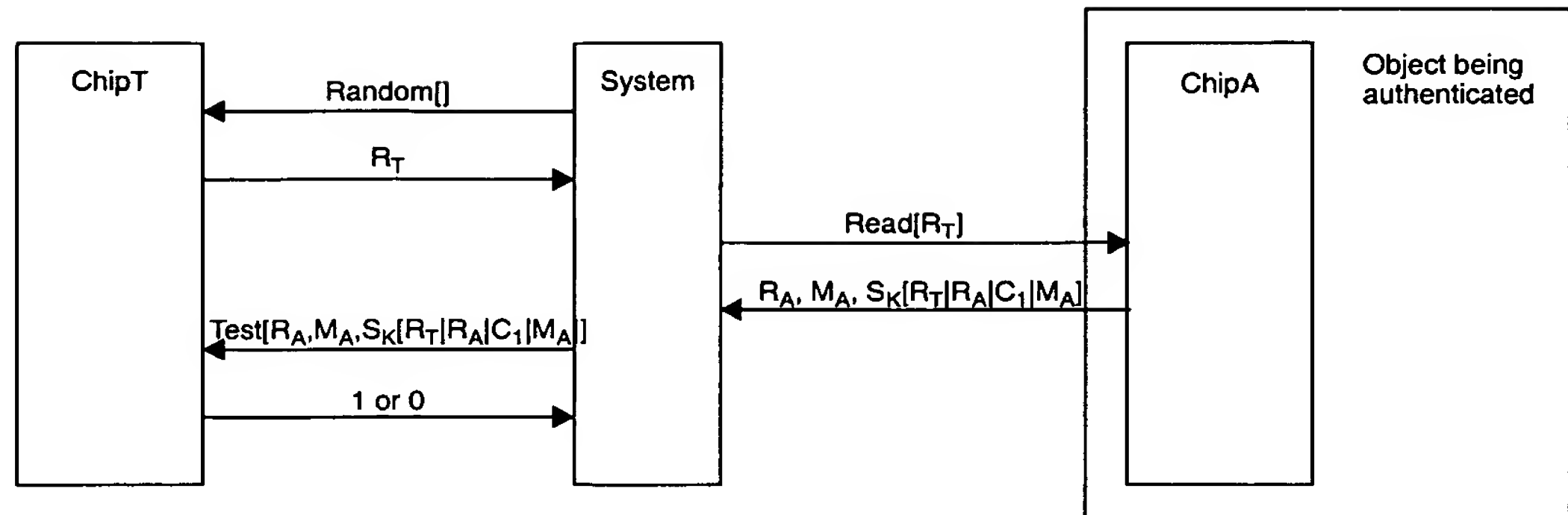


FIG. 334

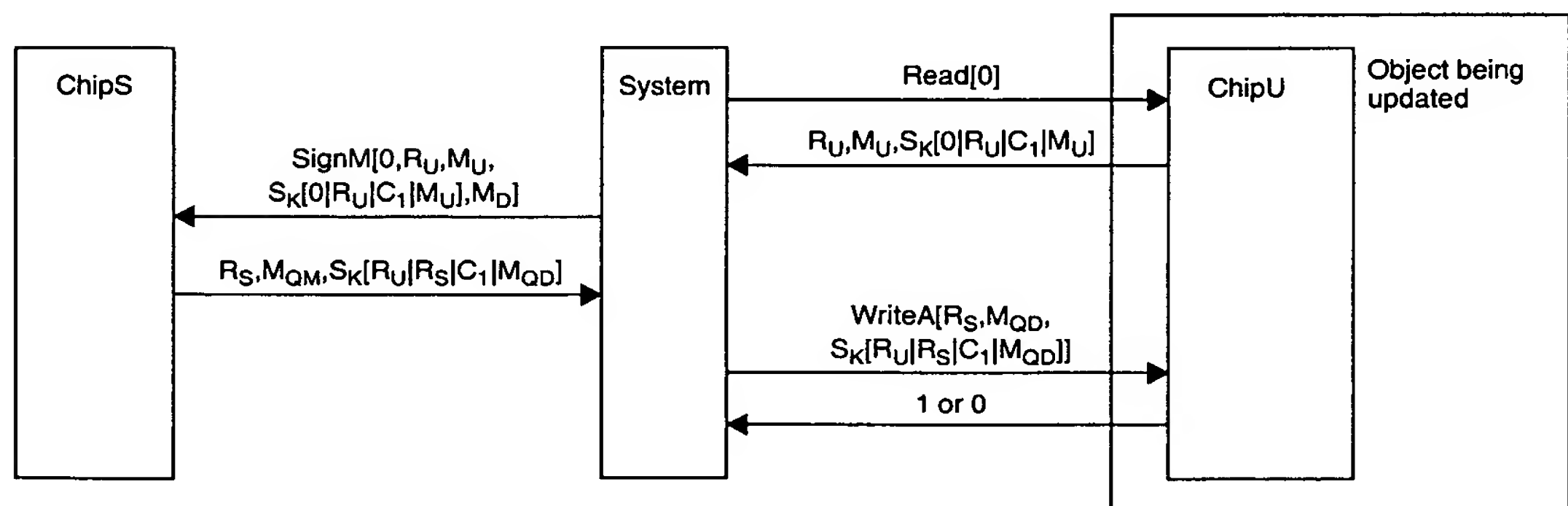


FIG. 335

284/331

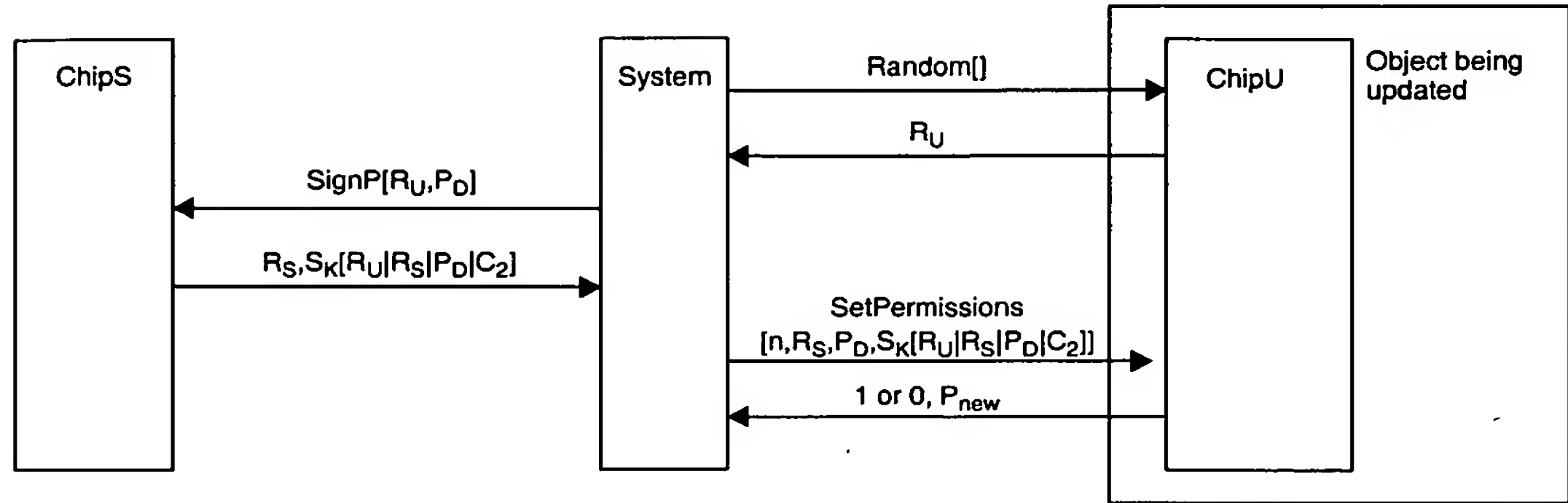


FIG. 336

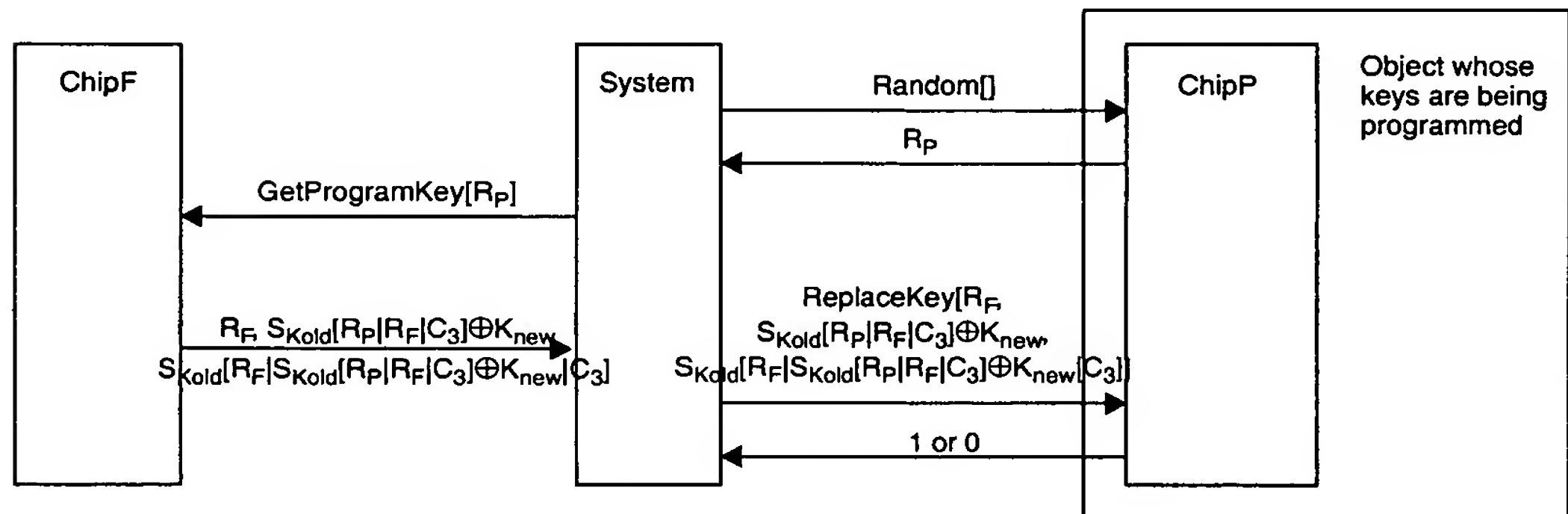


FIG. 337

285/331

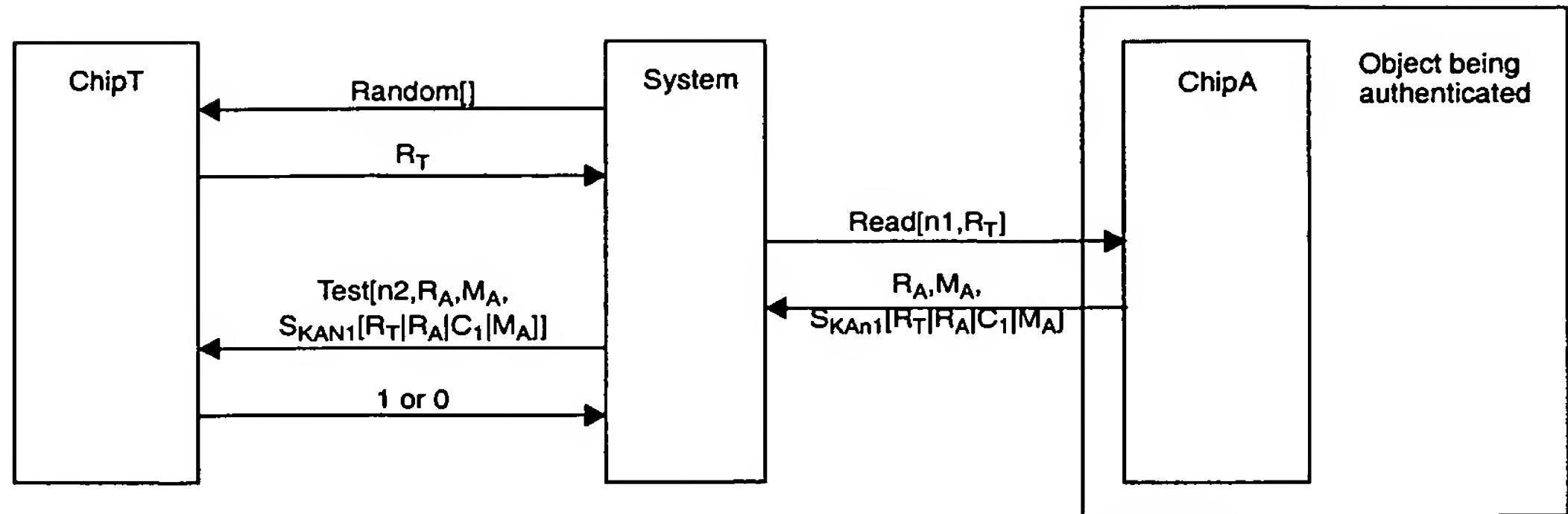


FIG. 338

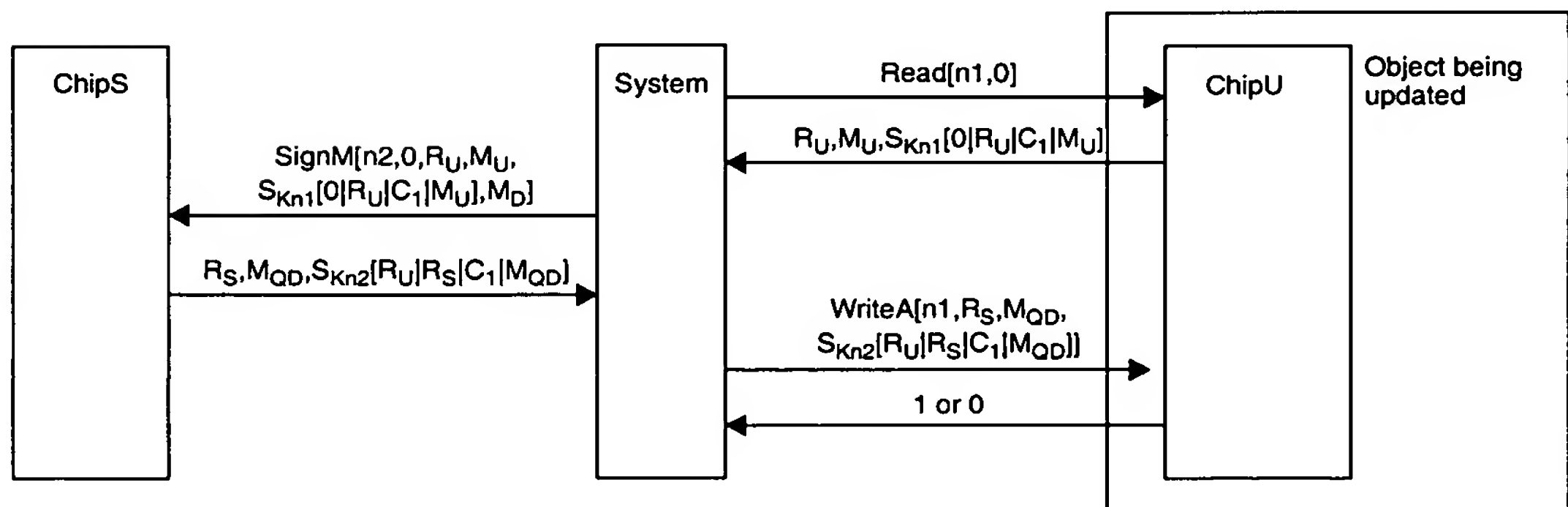


FIG. 339

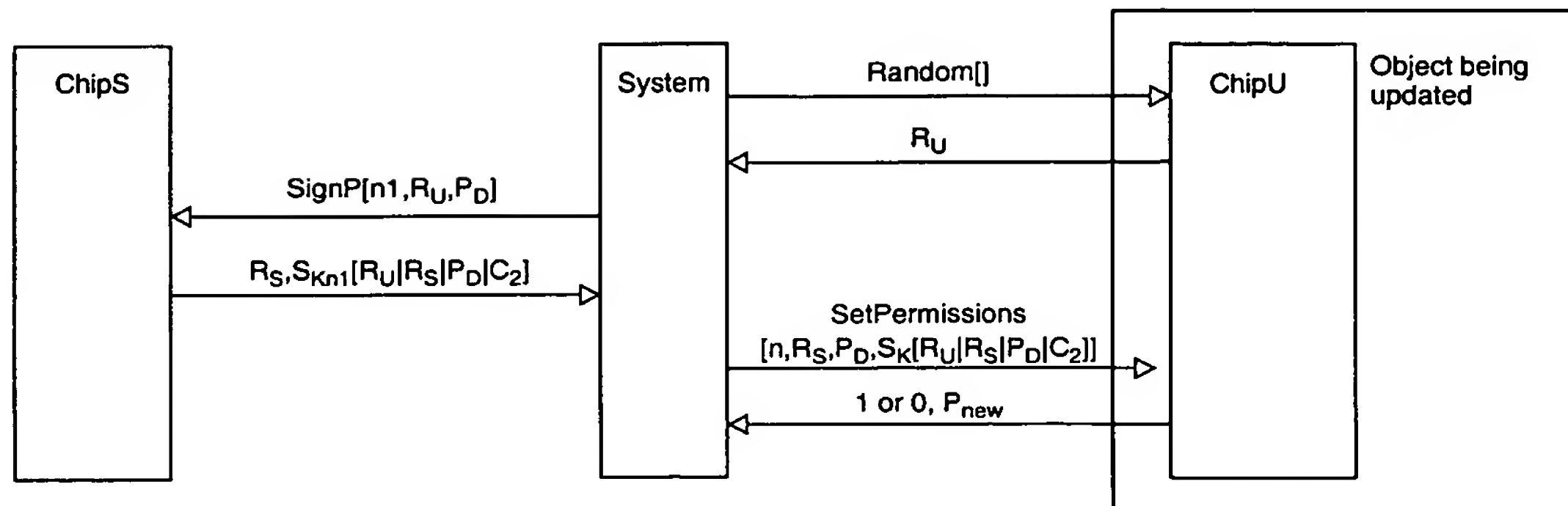


FIG. 340

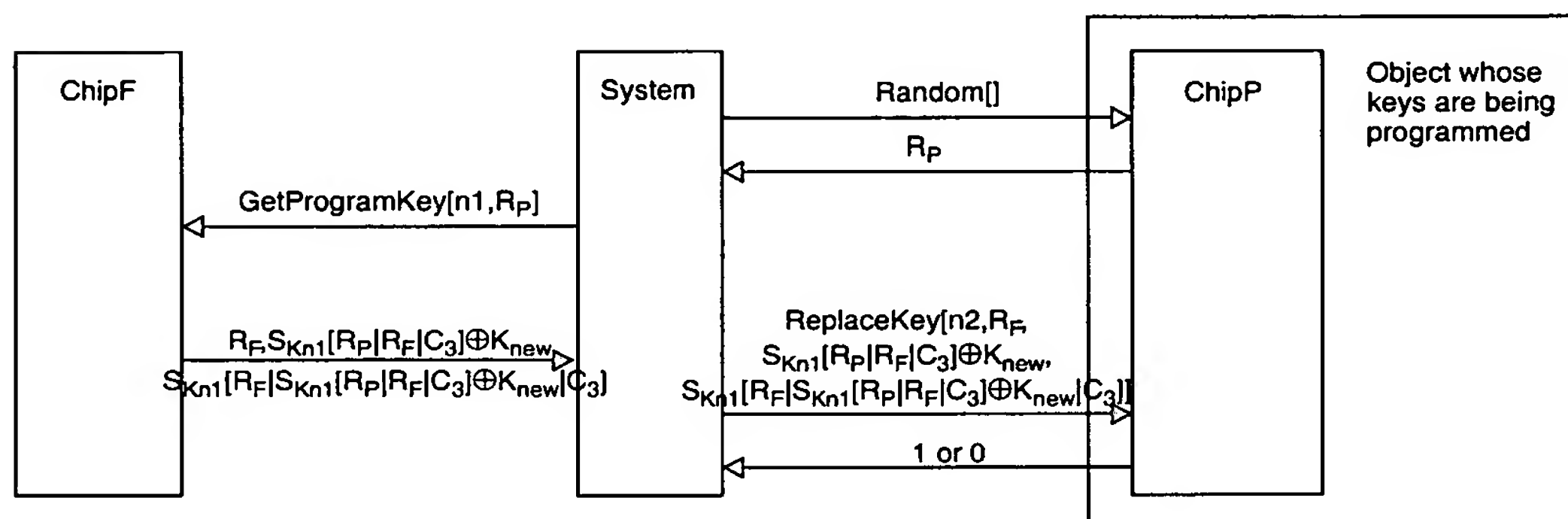


FIG. 341

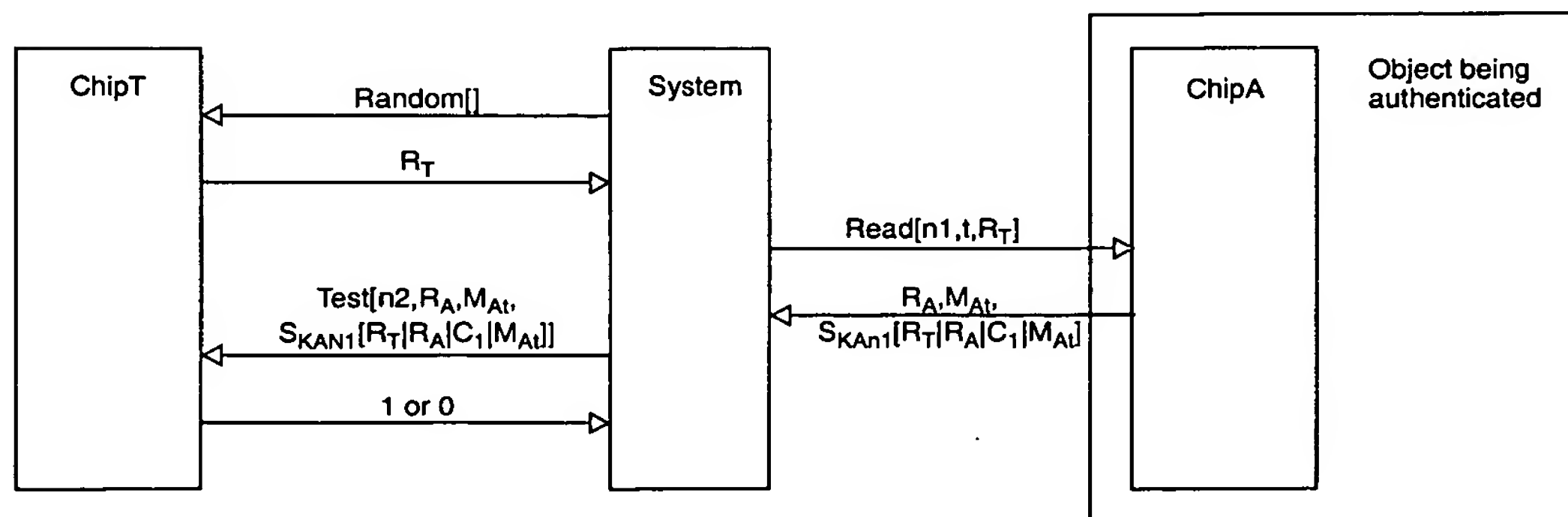


FIG. 342

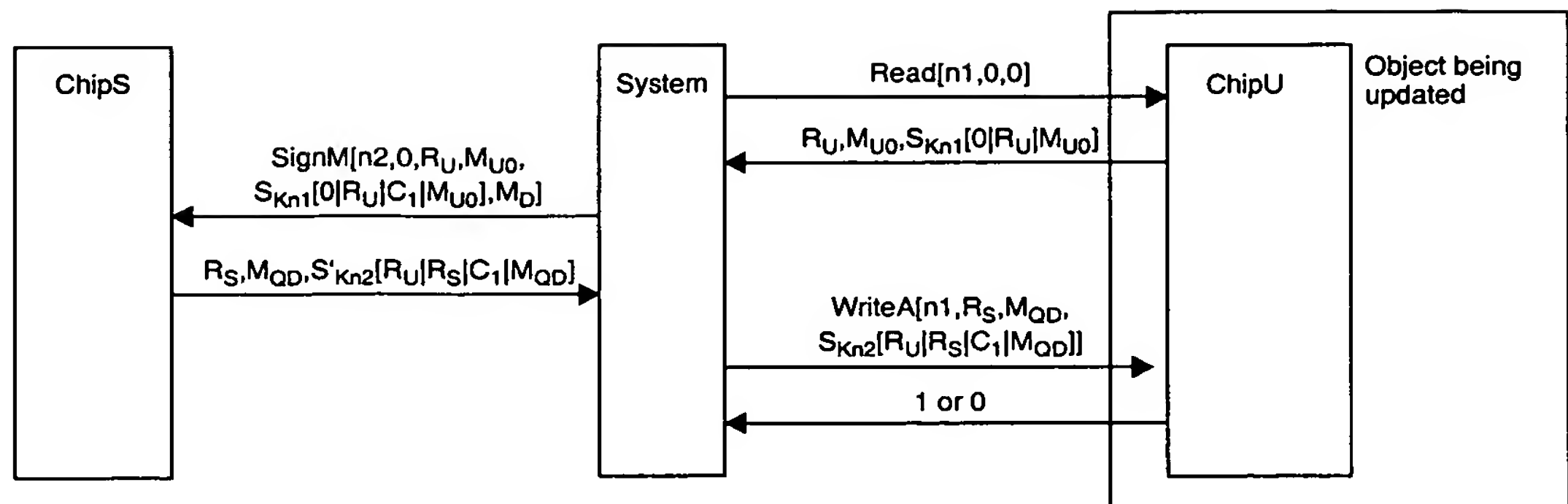


FIG. 343

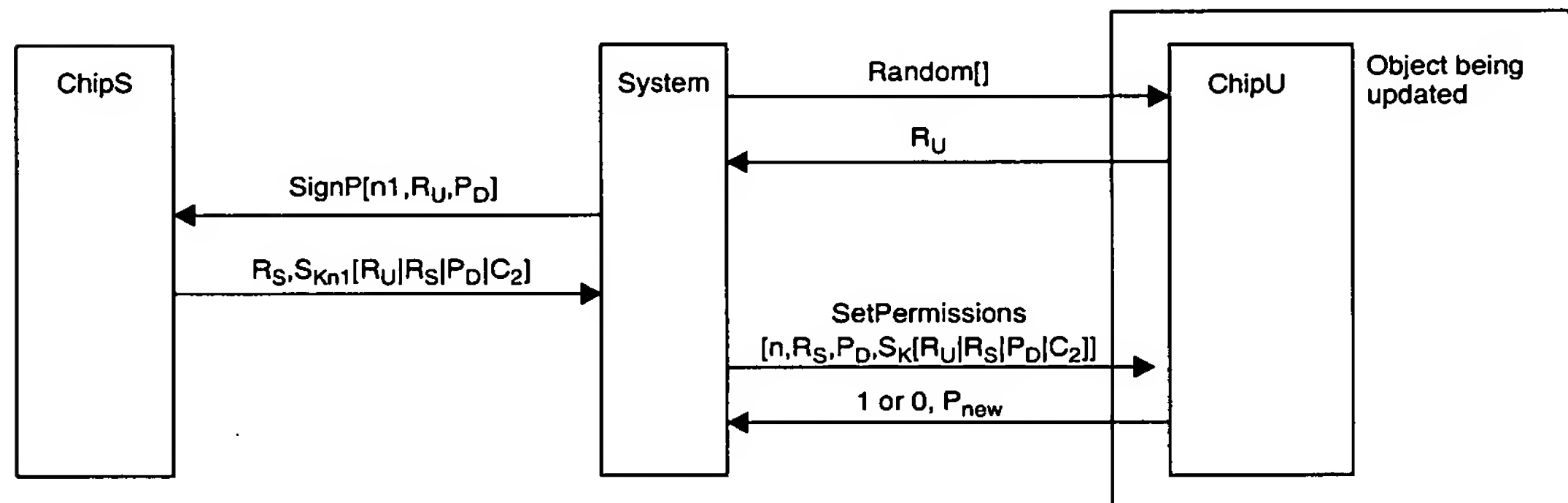


FIG. 344

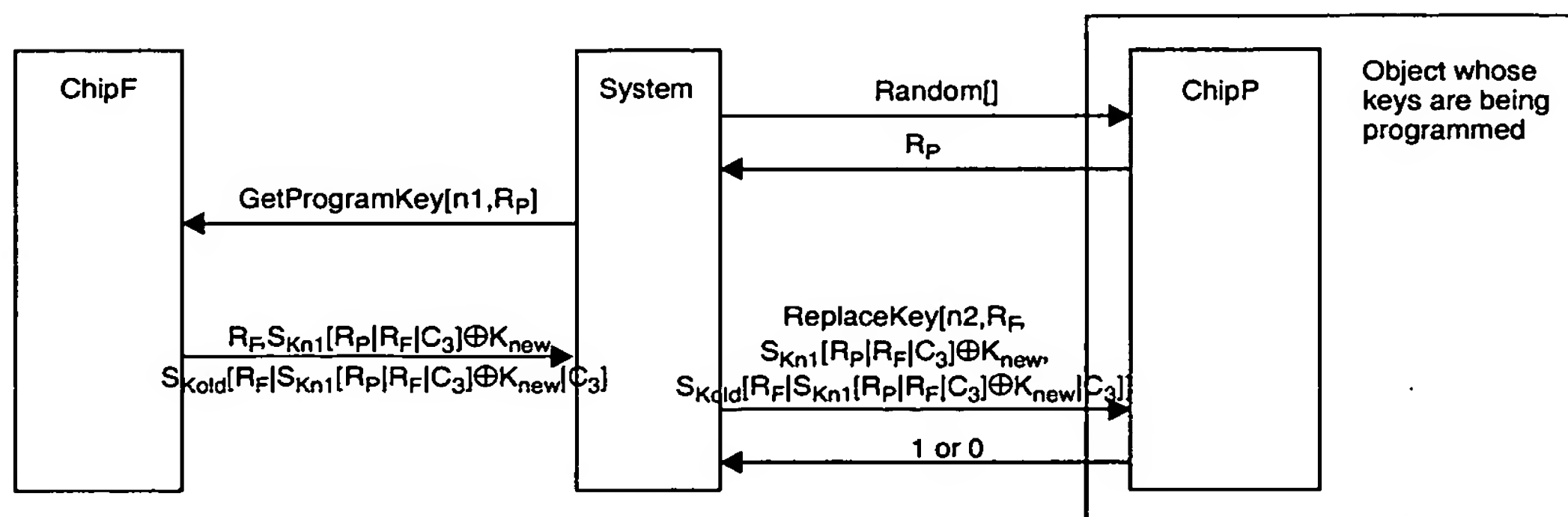


FIG. 345

288/331

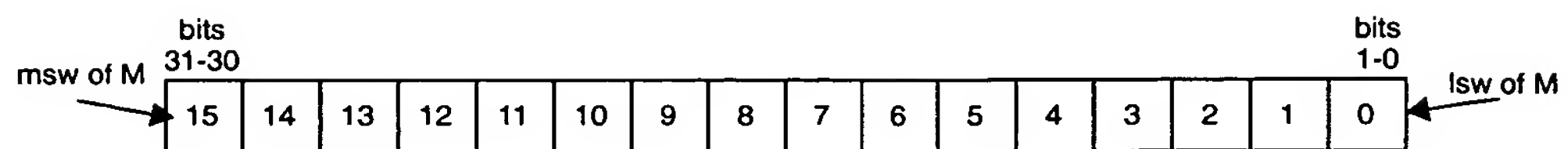


FIG. 346

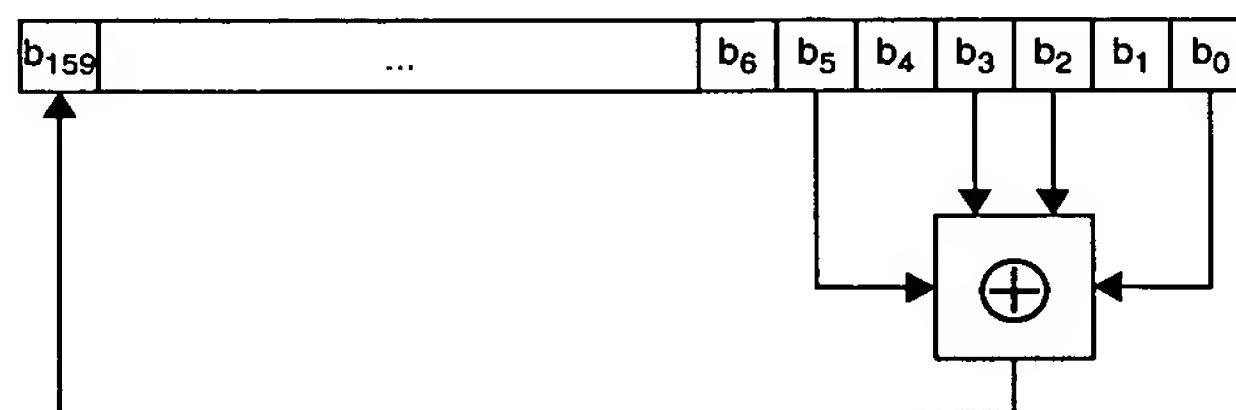


FIG. 347

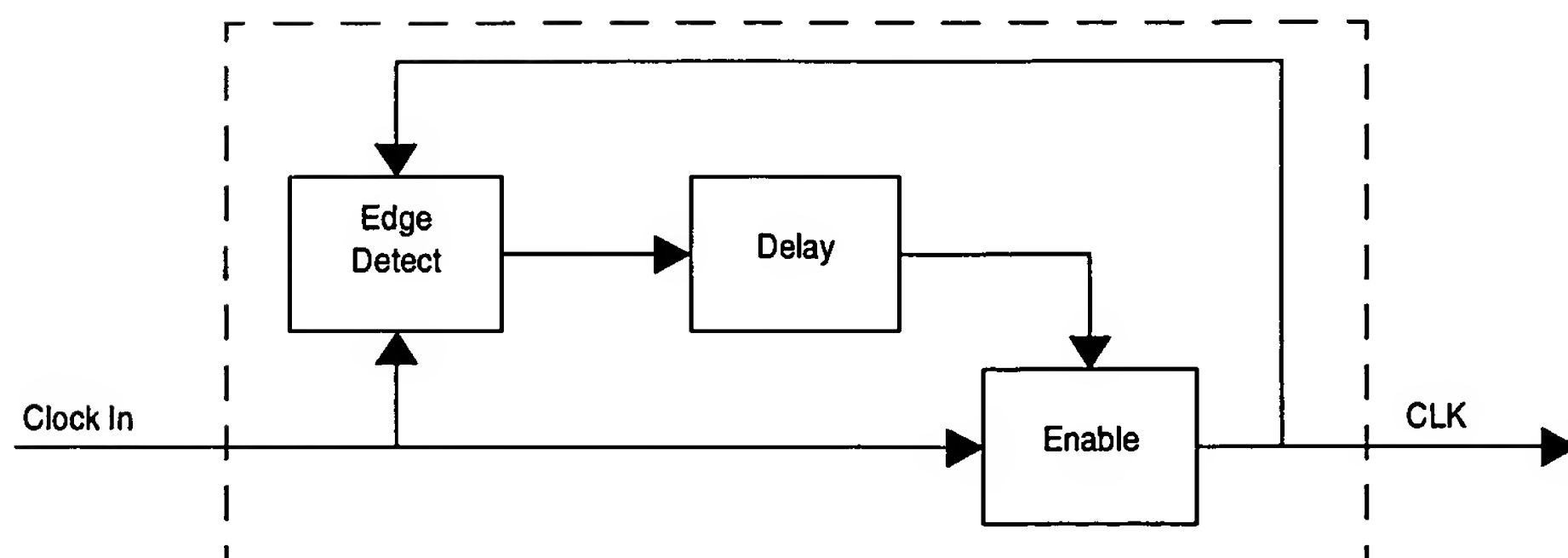


FIG. 348

289/331

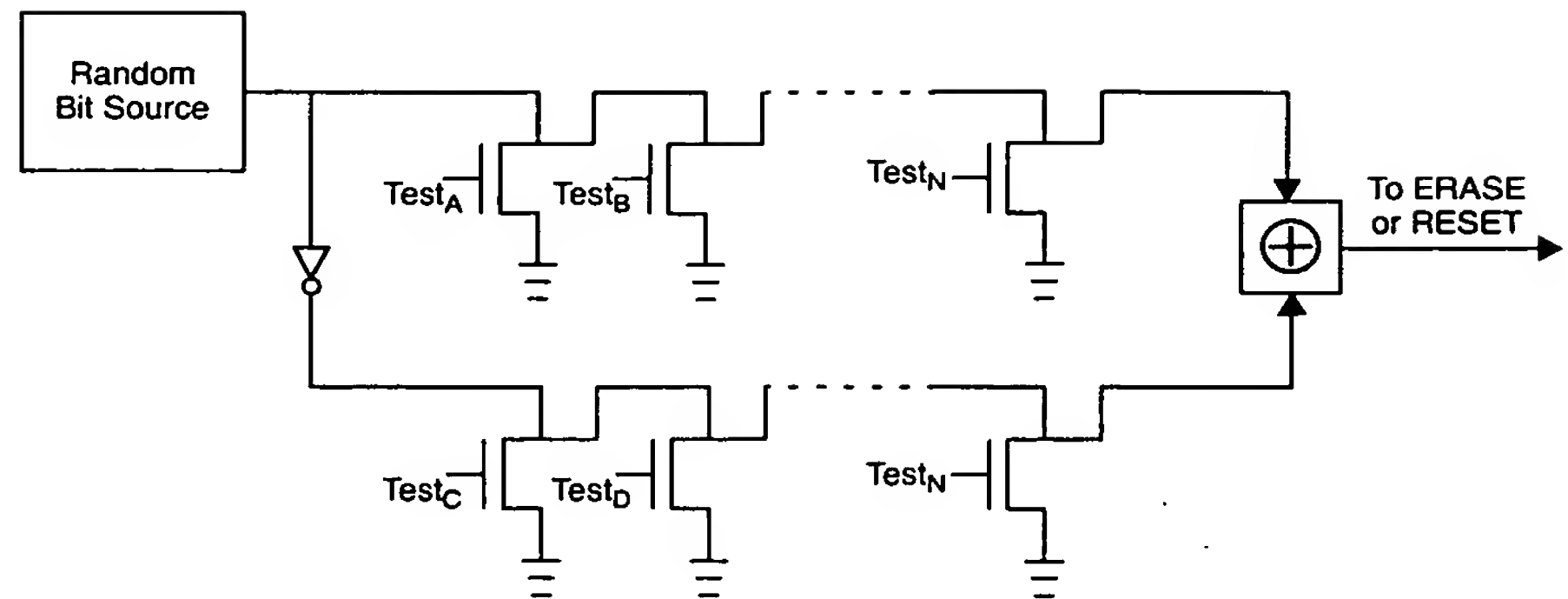


FIG. 349

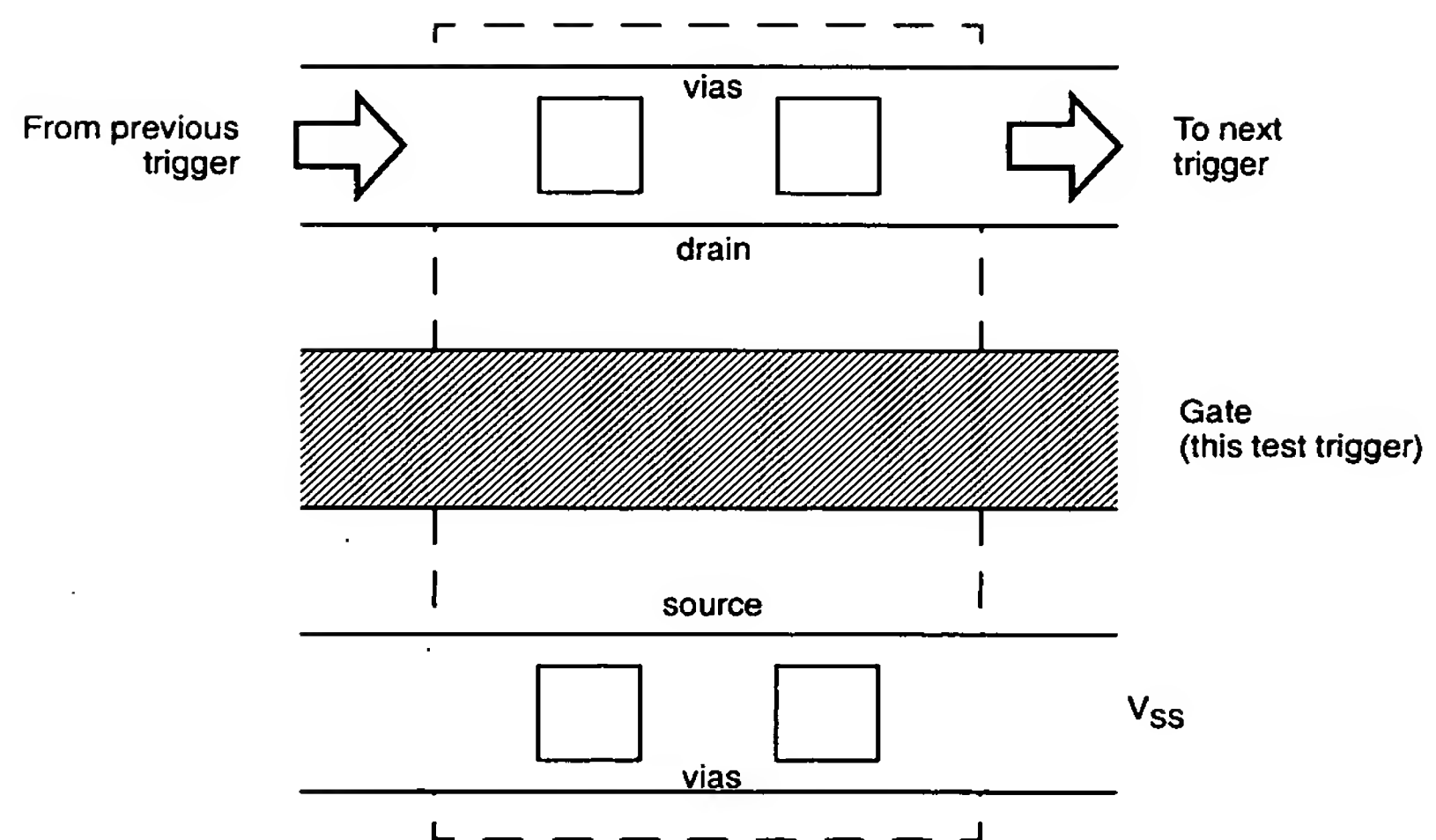


FIG. 350

290/331

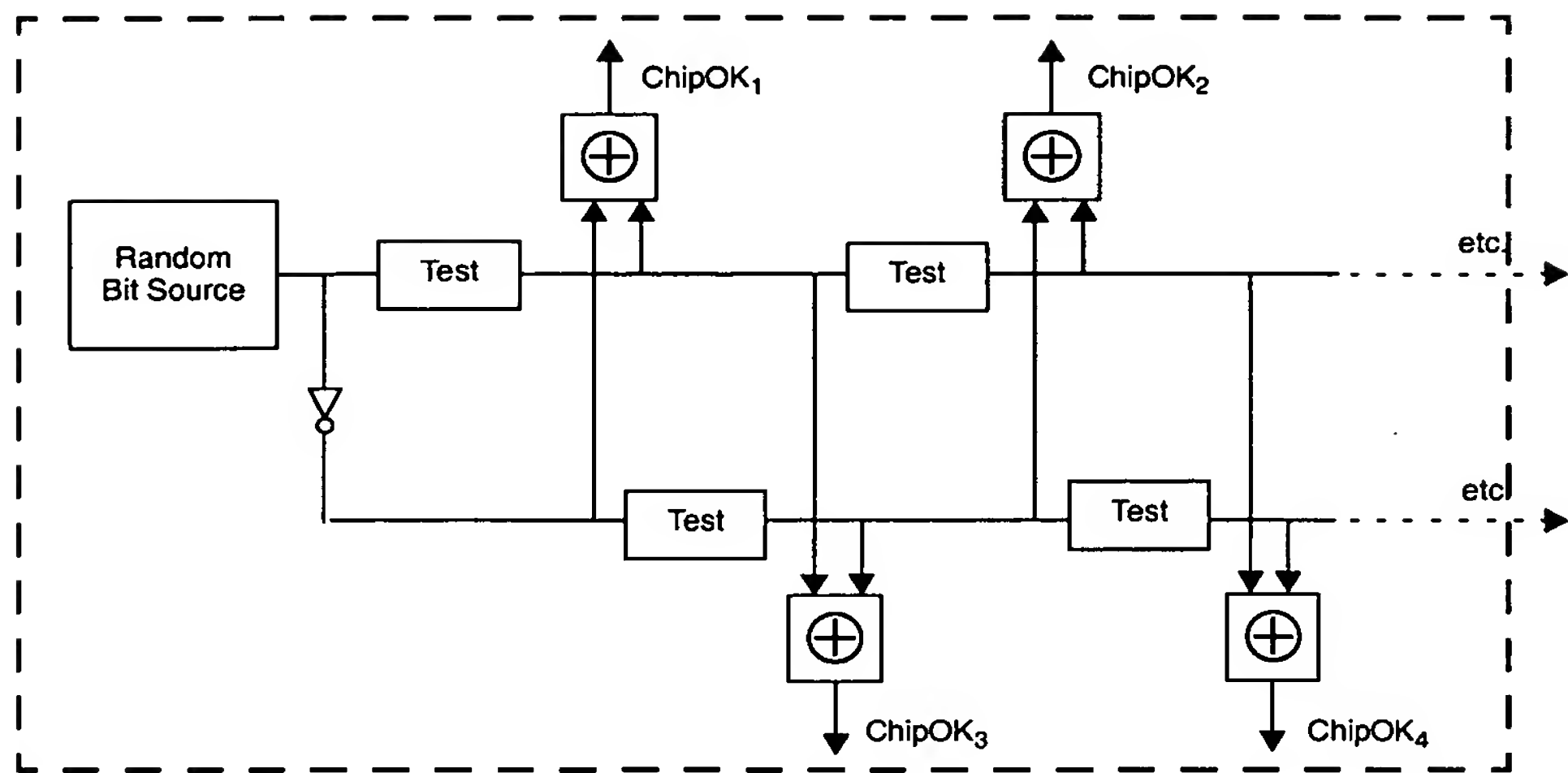


FIG. 351

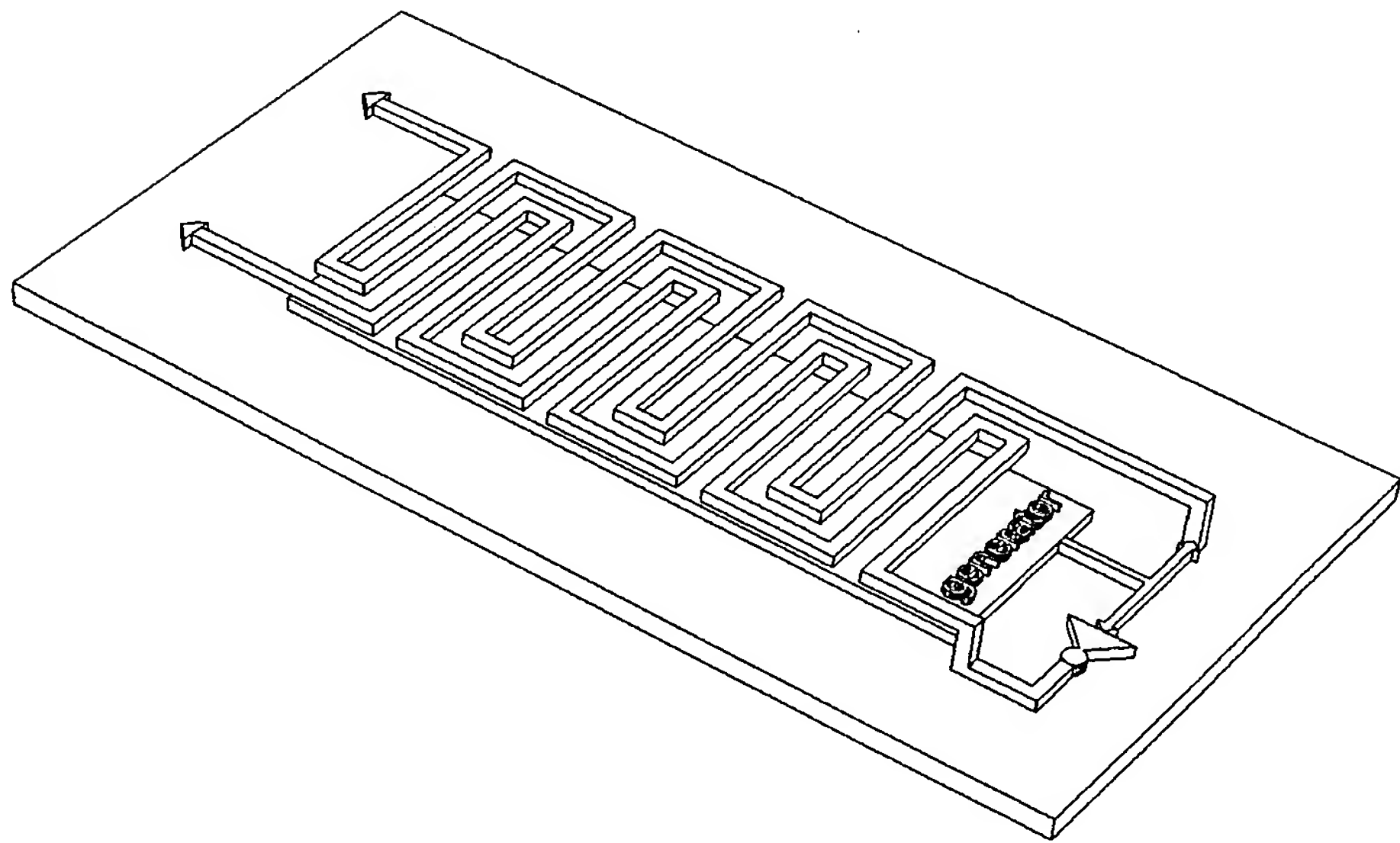


FIG. 352

291/331

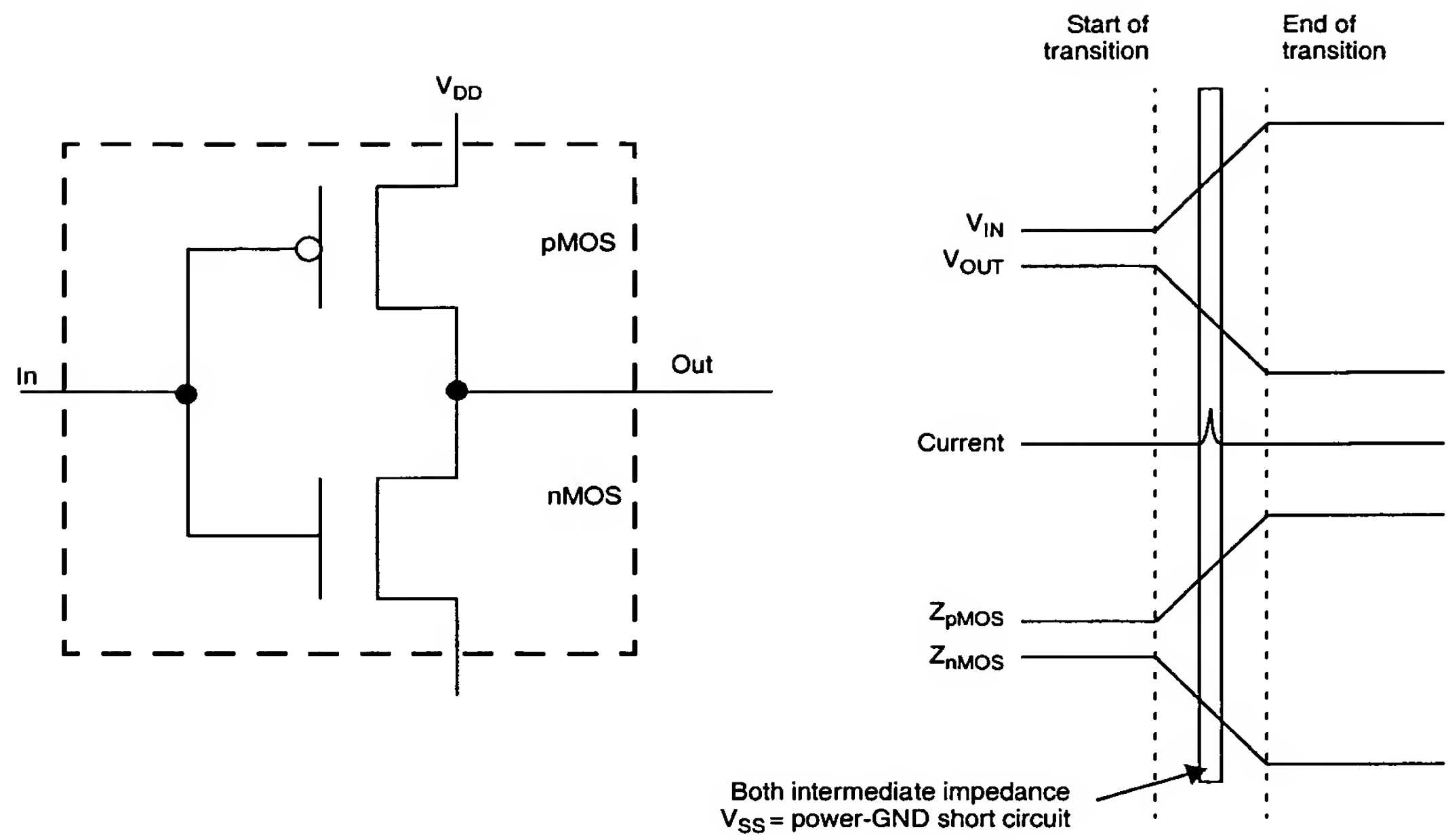


FIG. 353

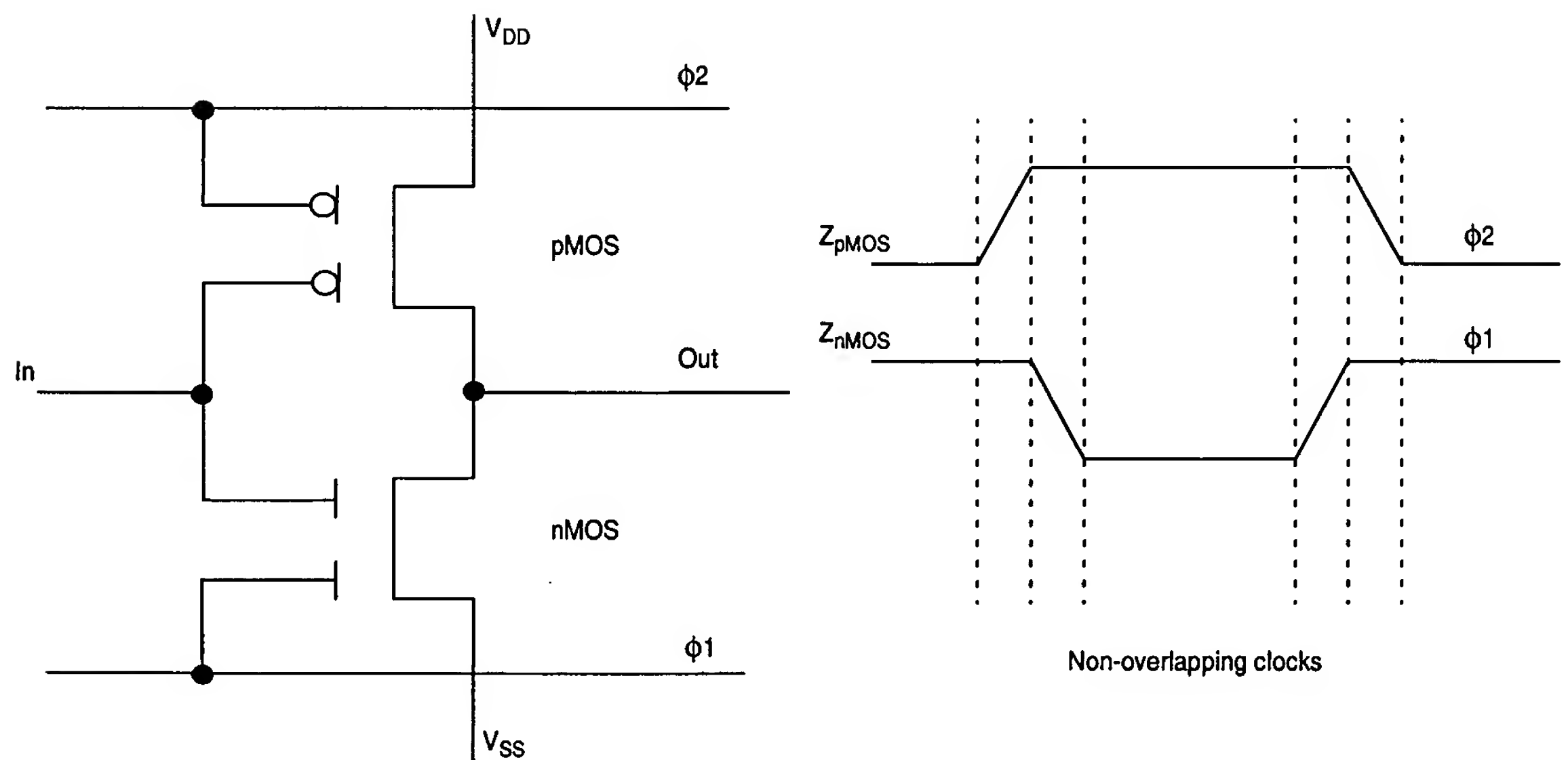


FIG. 354

292/331

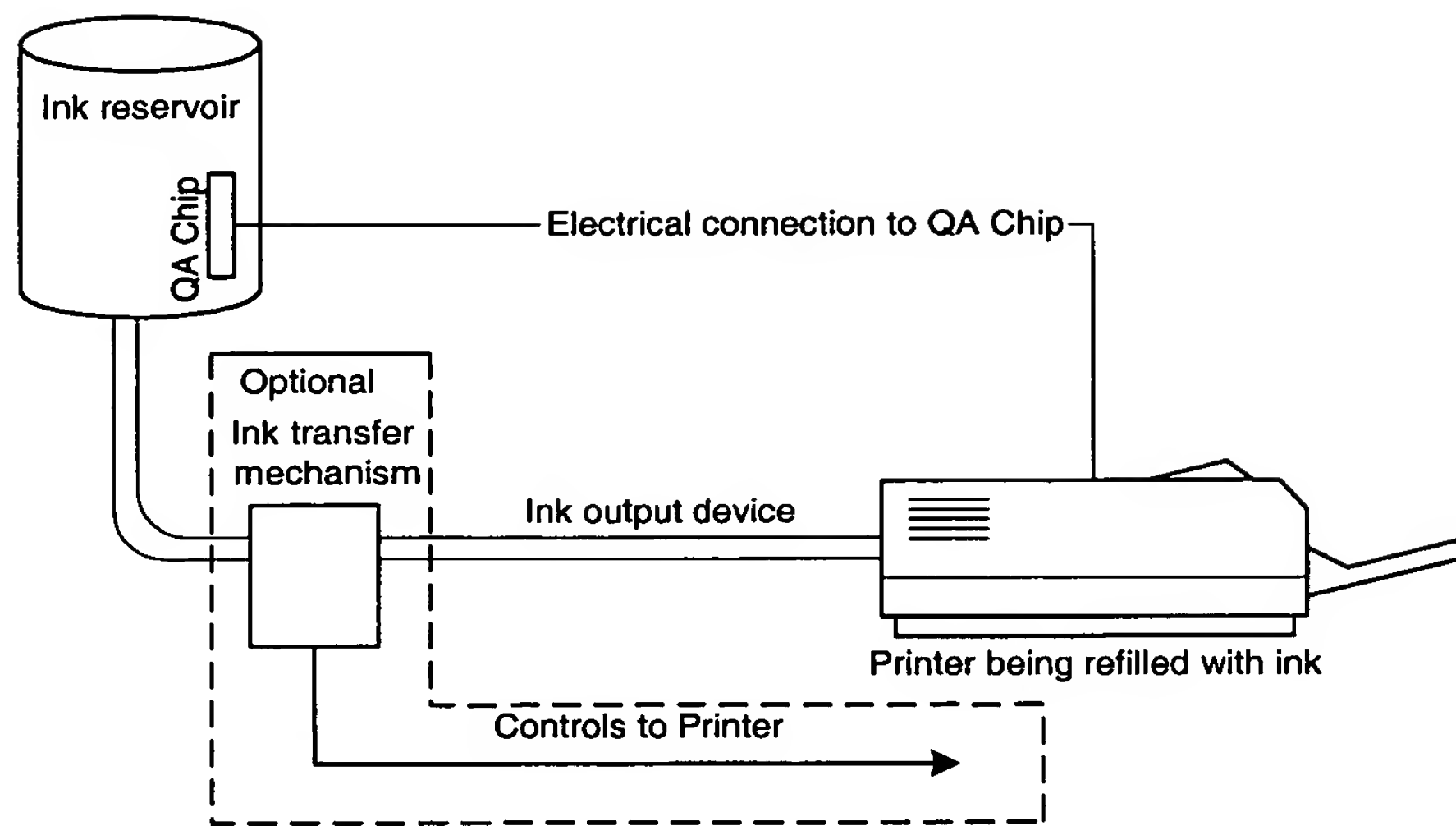


FIG. 355

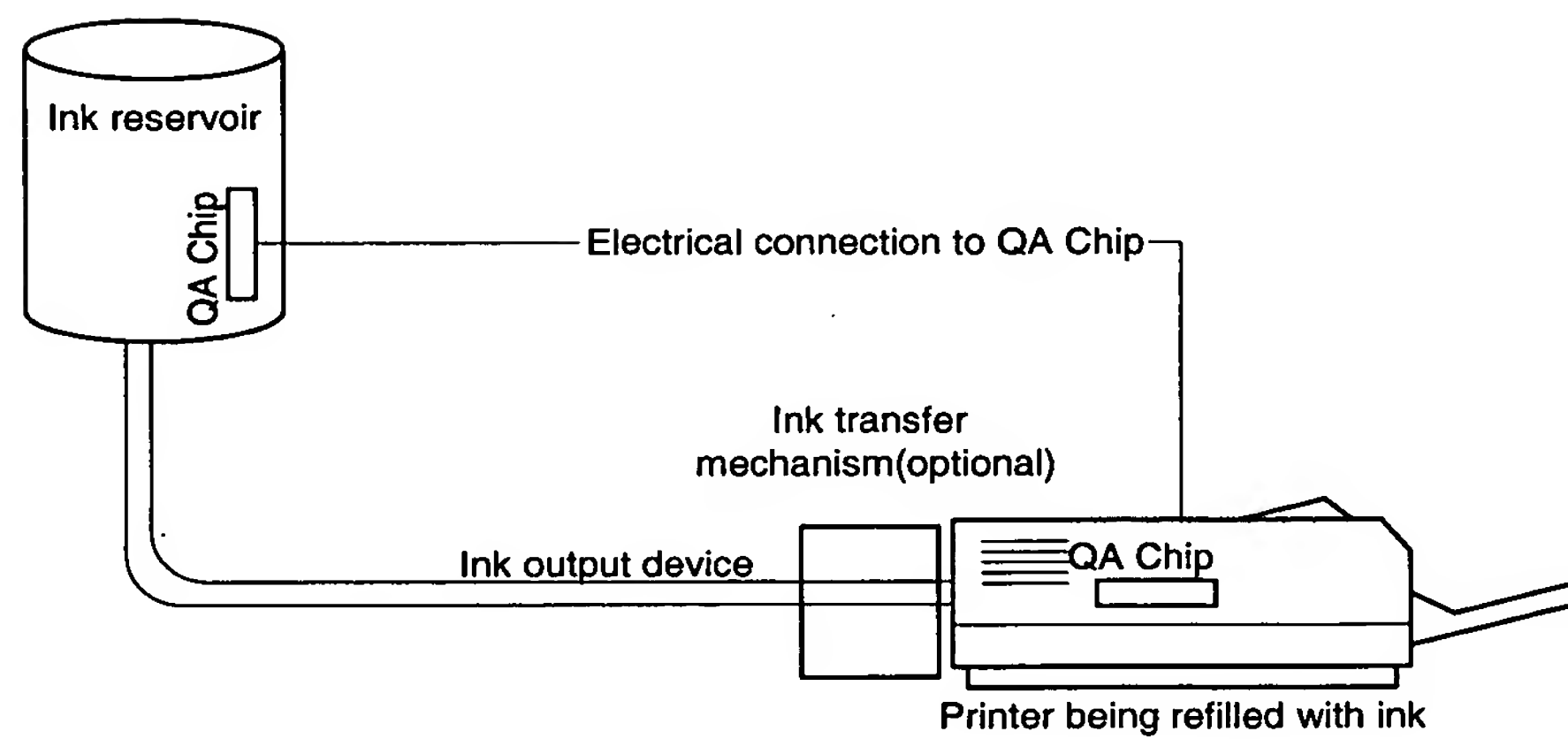


FIG. 356

293/331

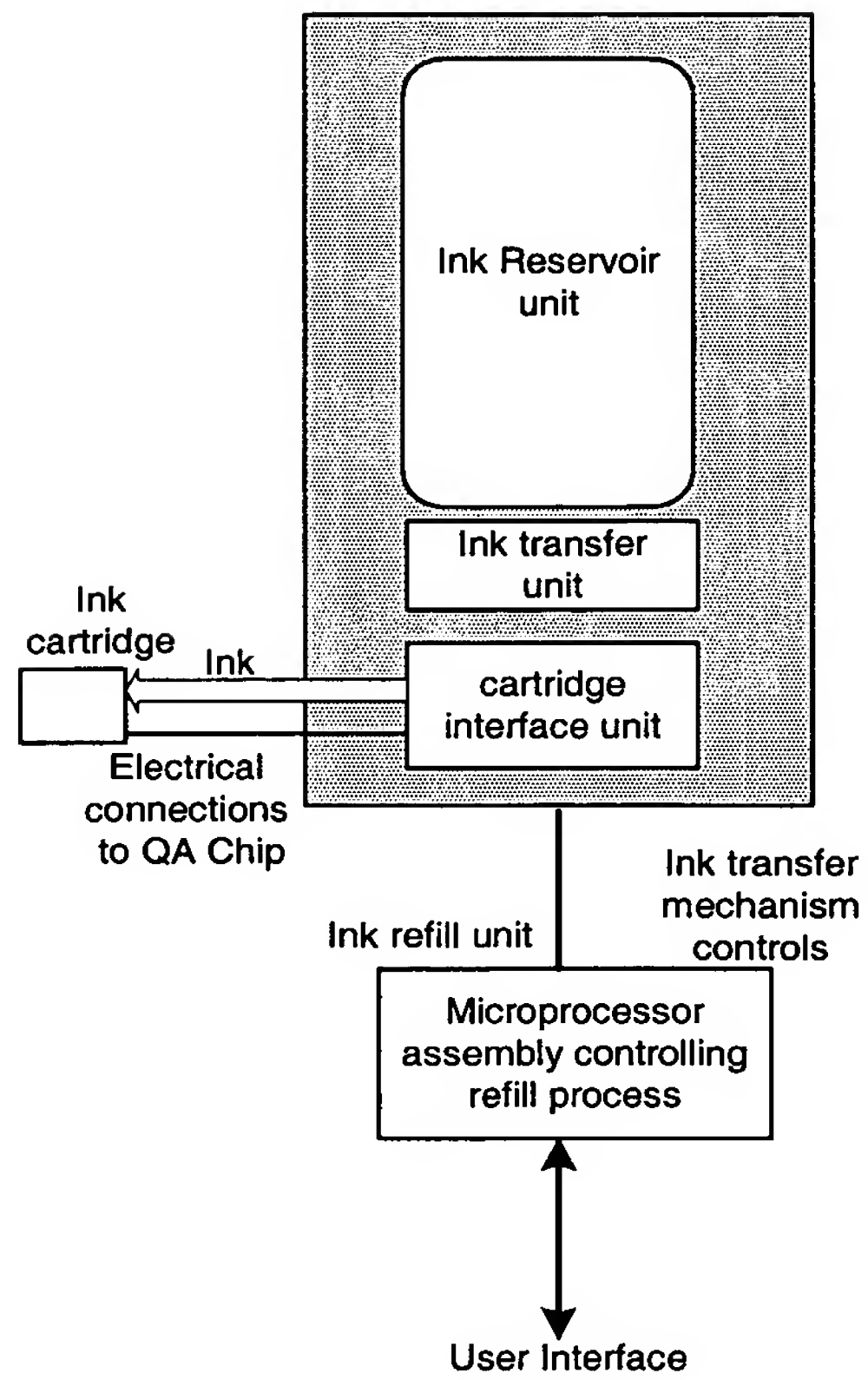


FIG. 357

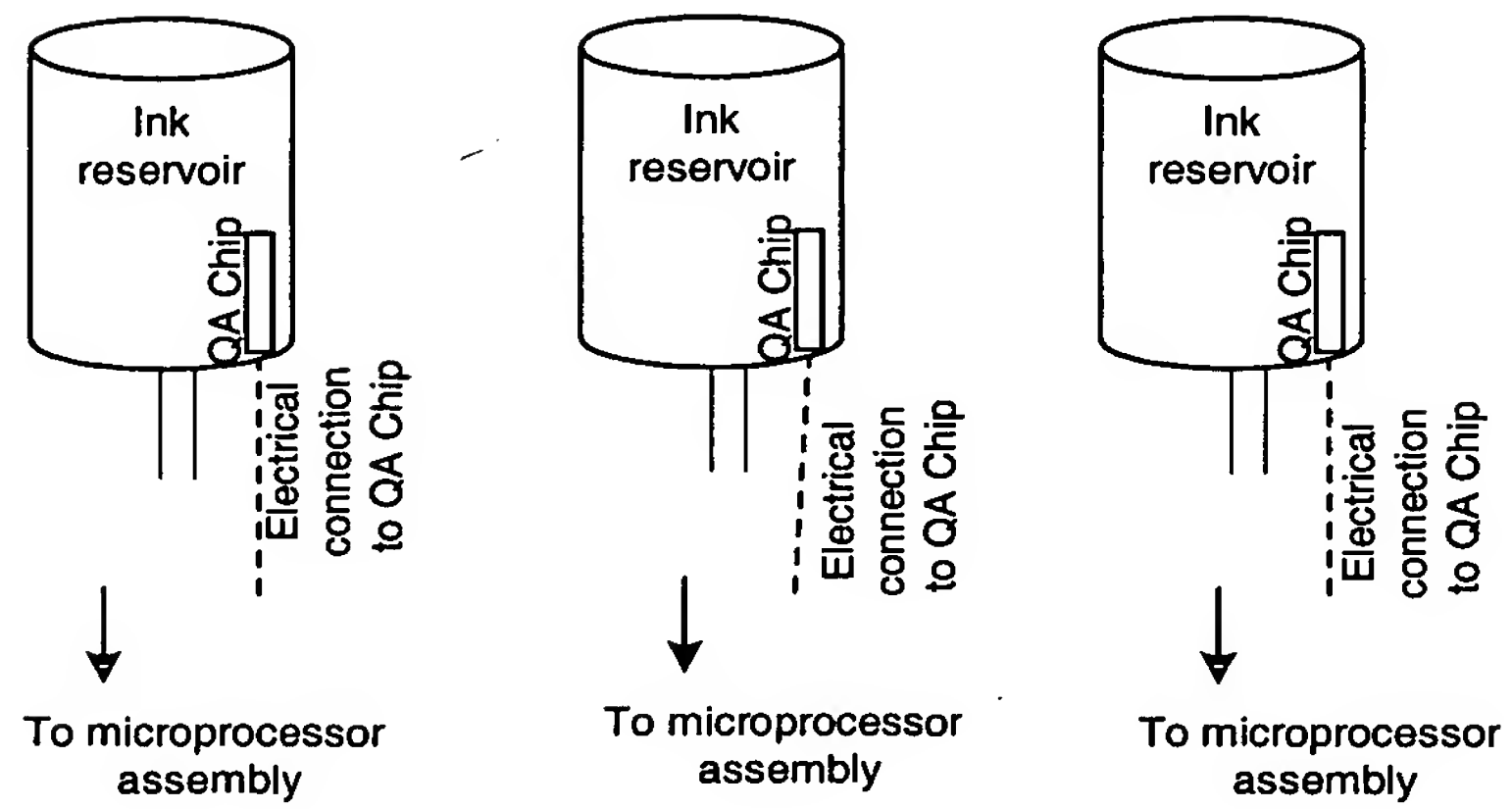


FIG. 358

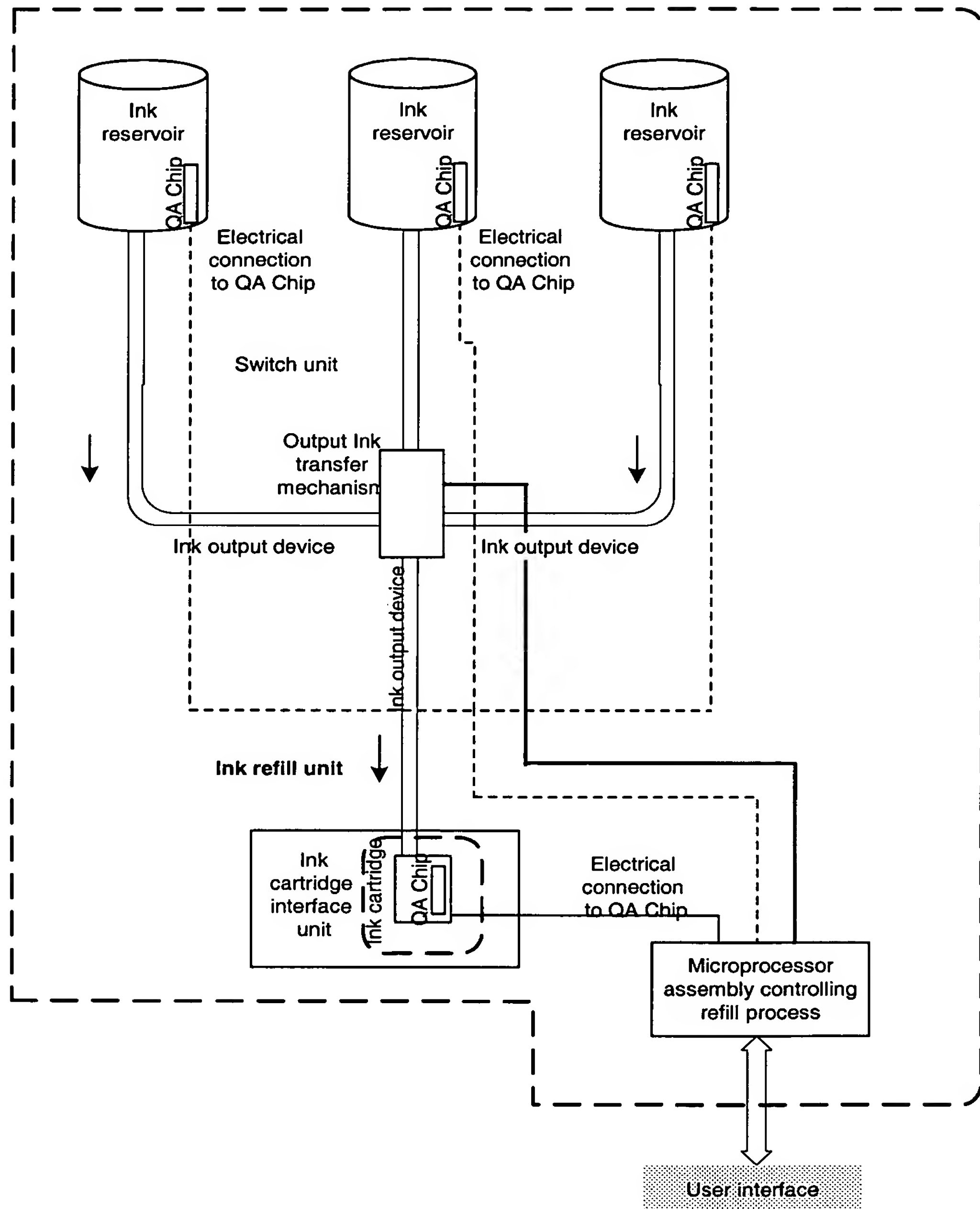


FIG. 359

295/331

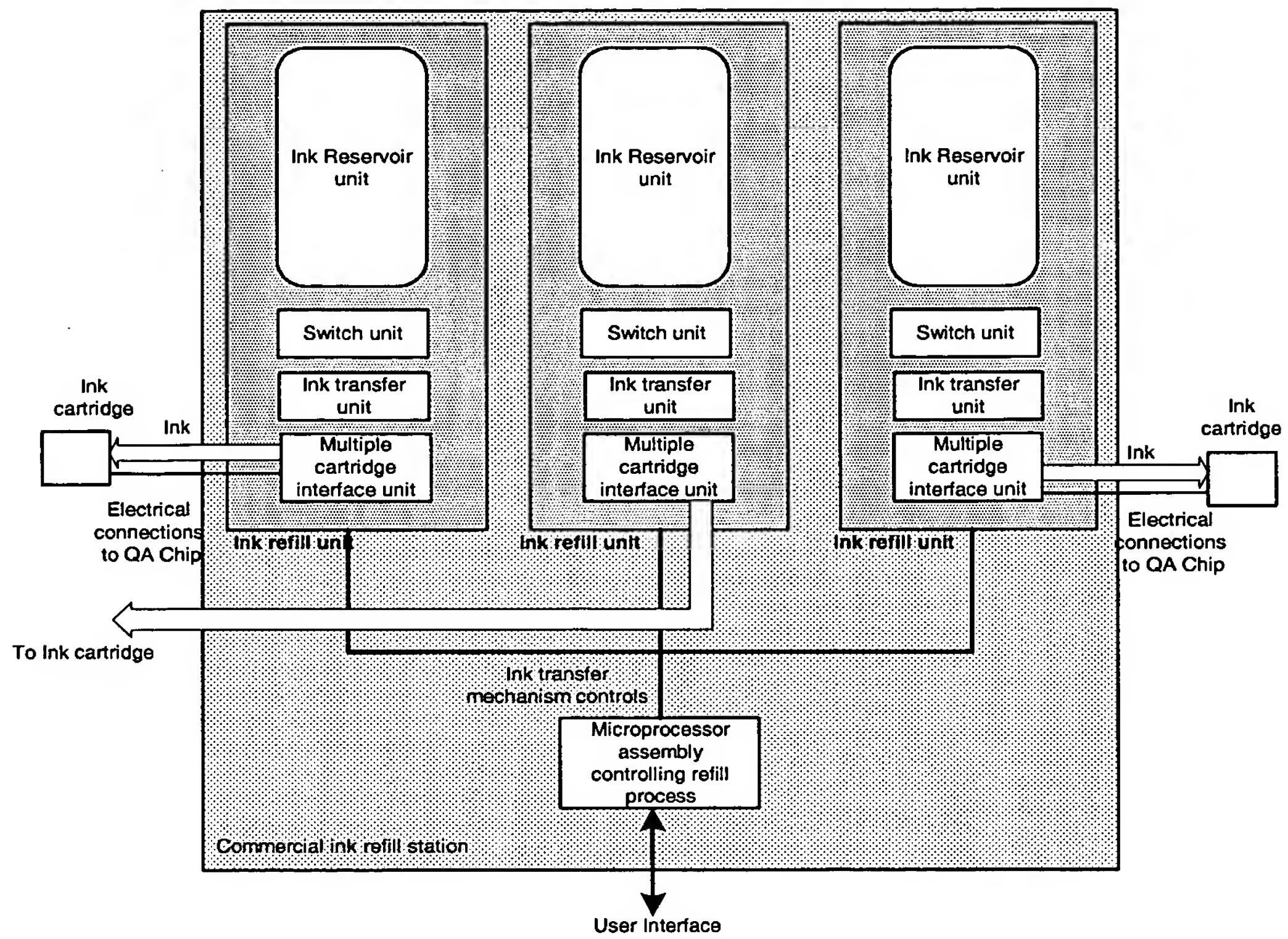


FIG. 360

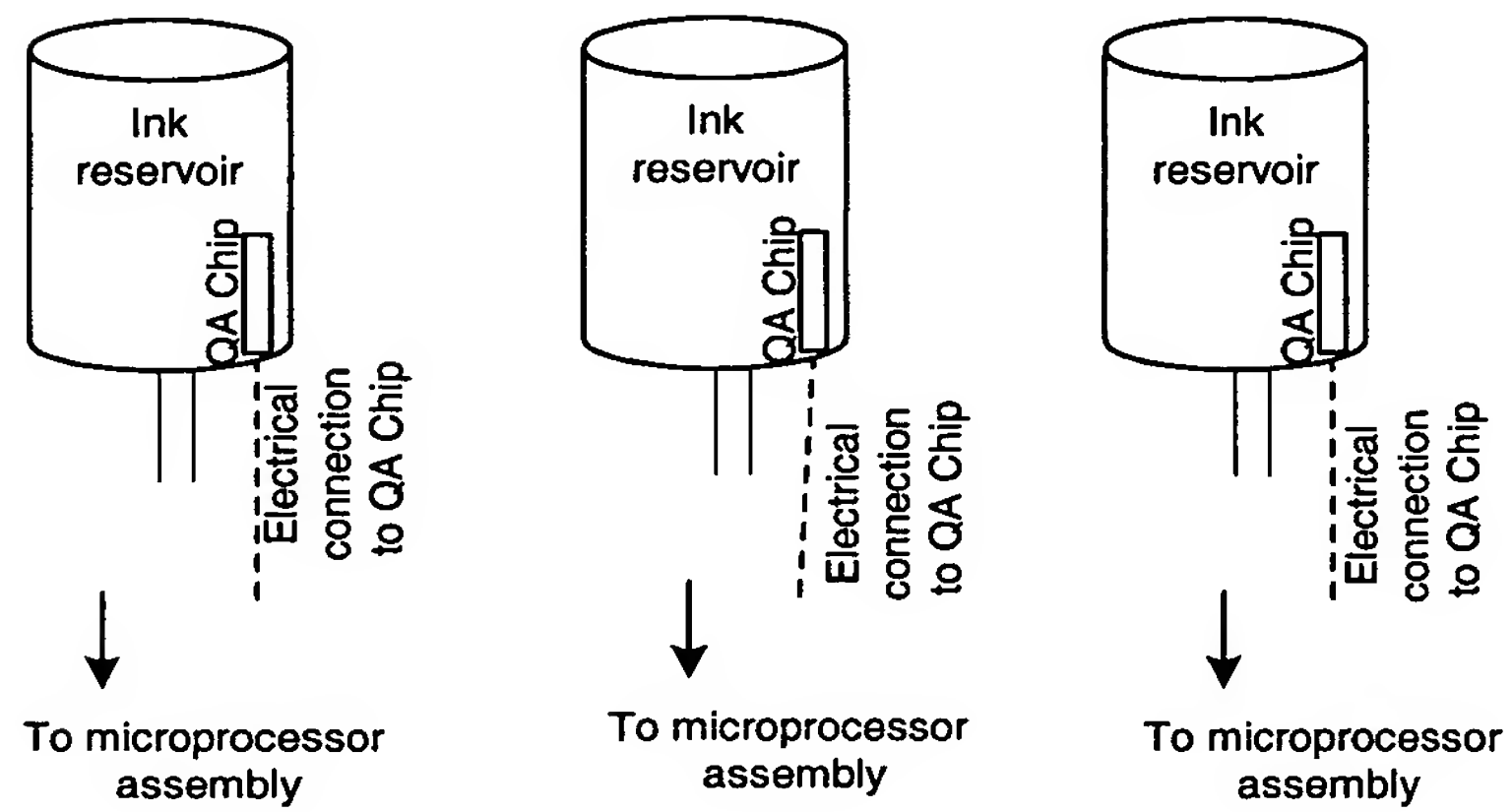


FIG. 361

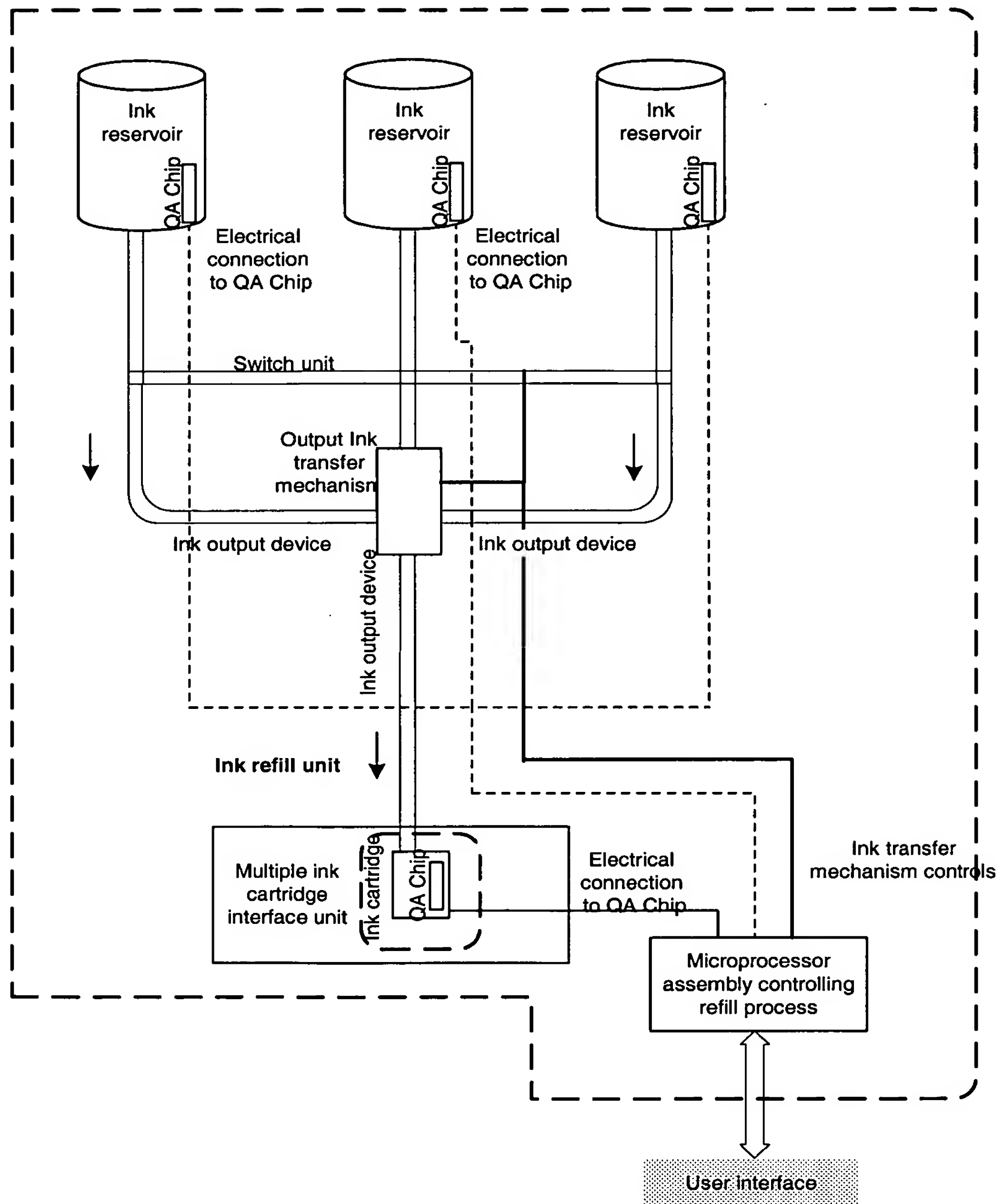


FIG. 362

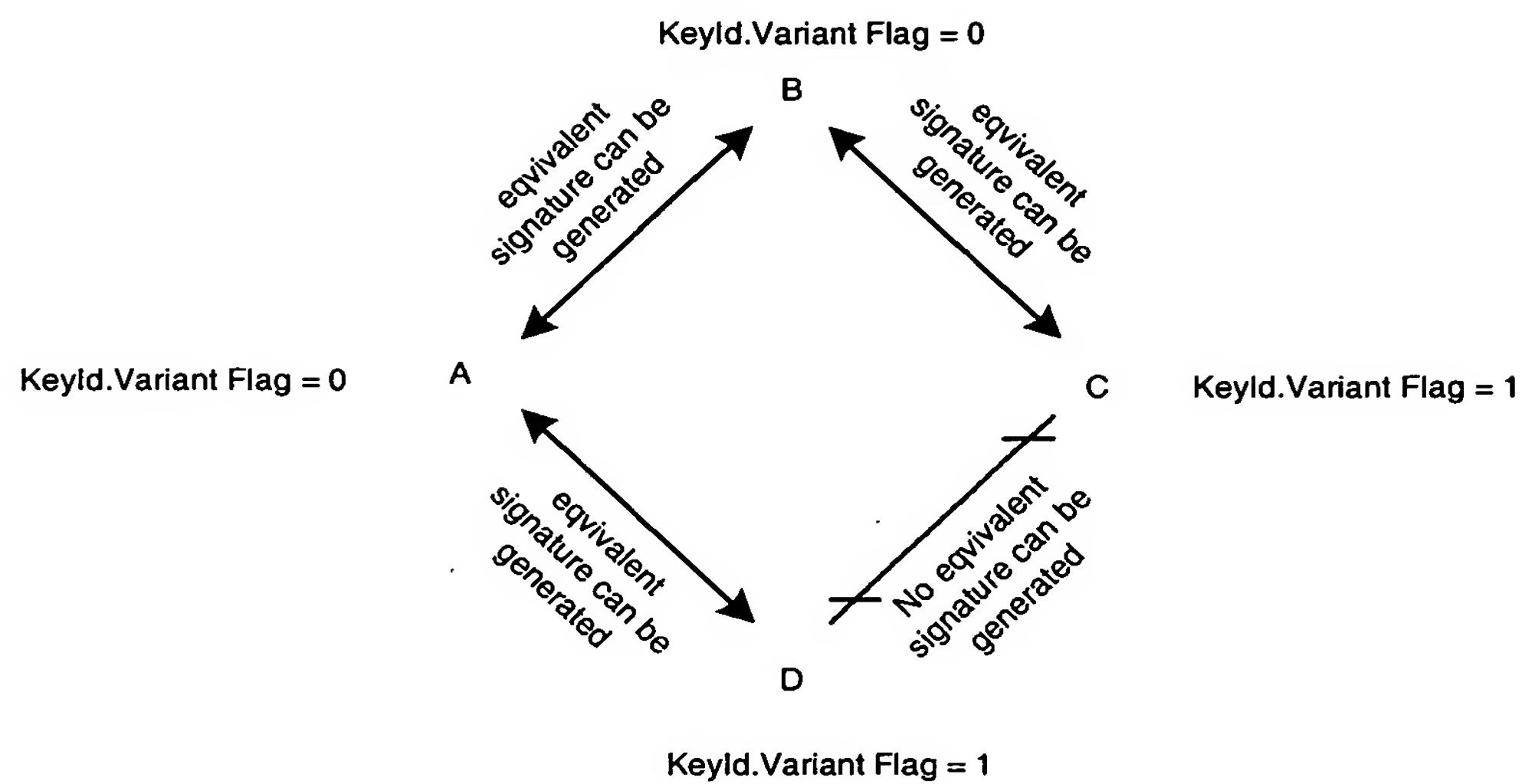


FIG. 363

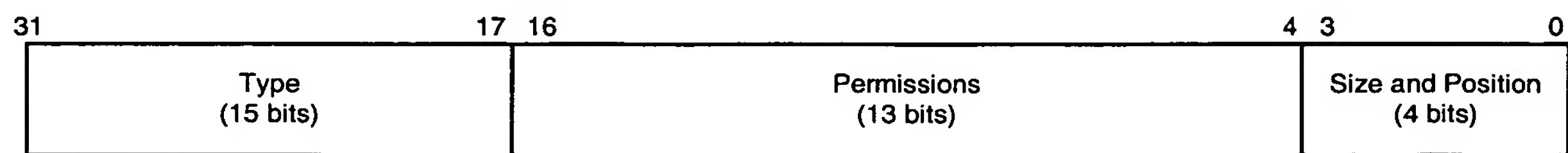


FIG. 364

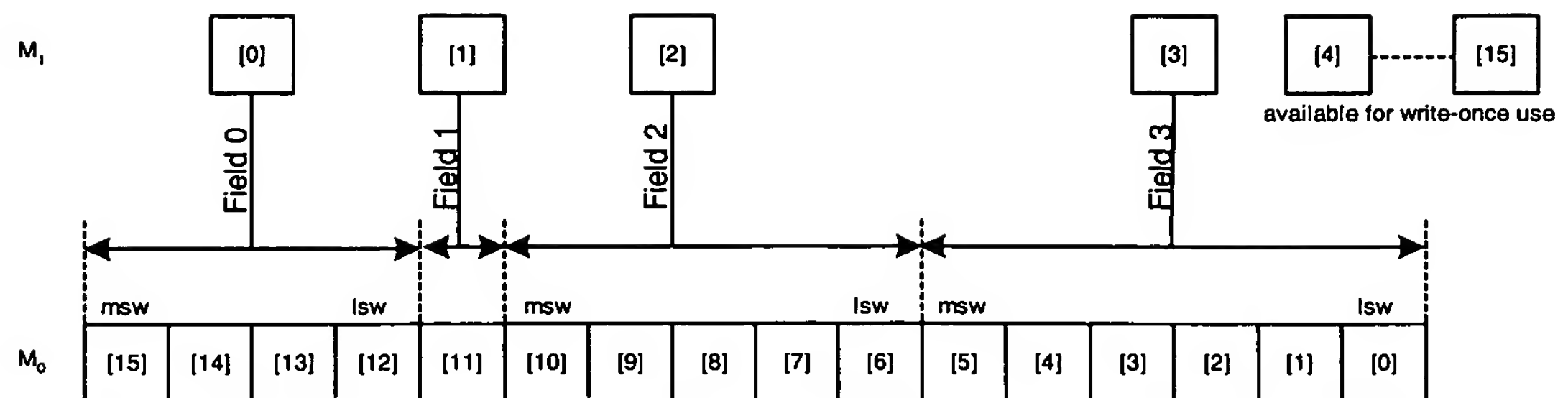


FIG. 365

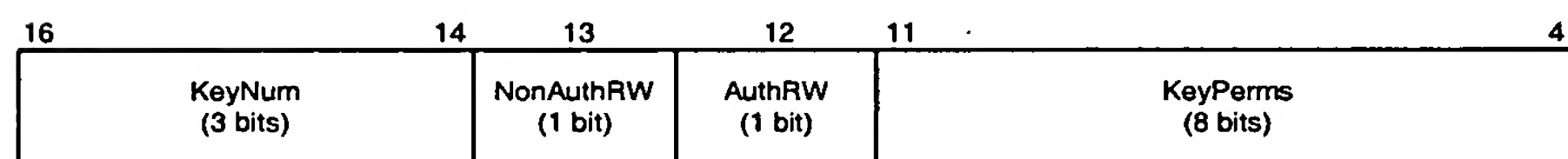


FIG. 366

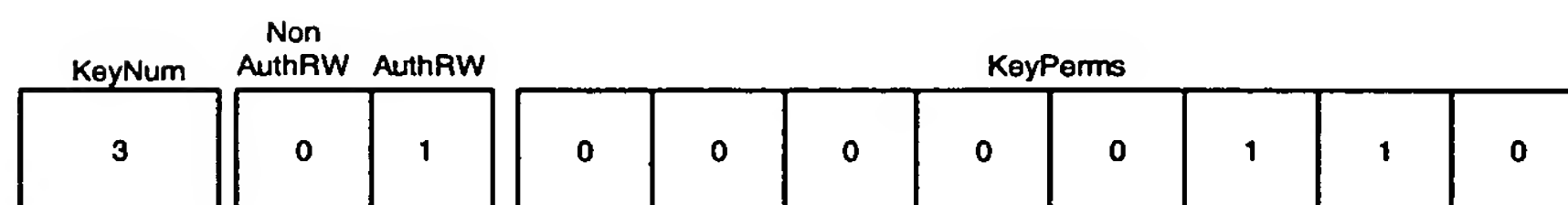


FIG. 367

299/331

KeyNum	Non AuthRW	AuthRW	KeyPerms							
3	1	1	0	0	0	0	1	1	1	0

FIG. 368

31	17	16	14	13	12	11	4	3	0
Type (15 bits)			KeyNum (3 bits)	NonAuth RW (1 bit)	AuthRW (1 bit)	KeyPerms (8 bits)		EndPos (4 bits)	

FIG. 369

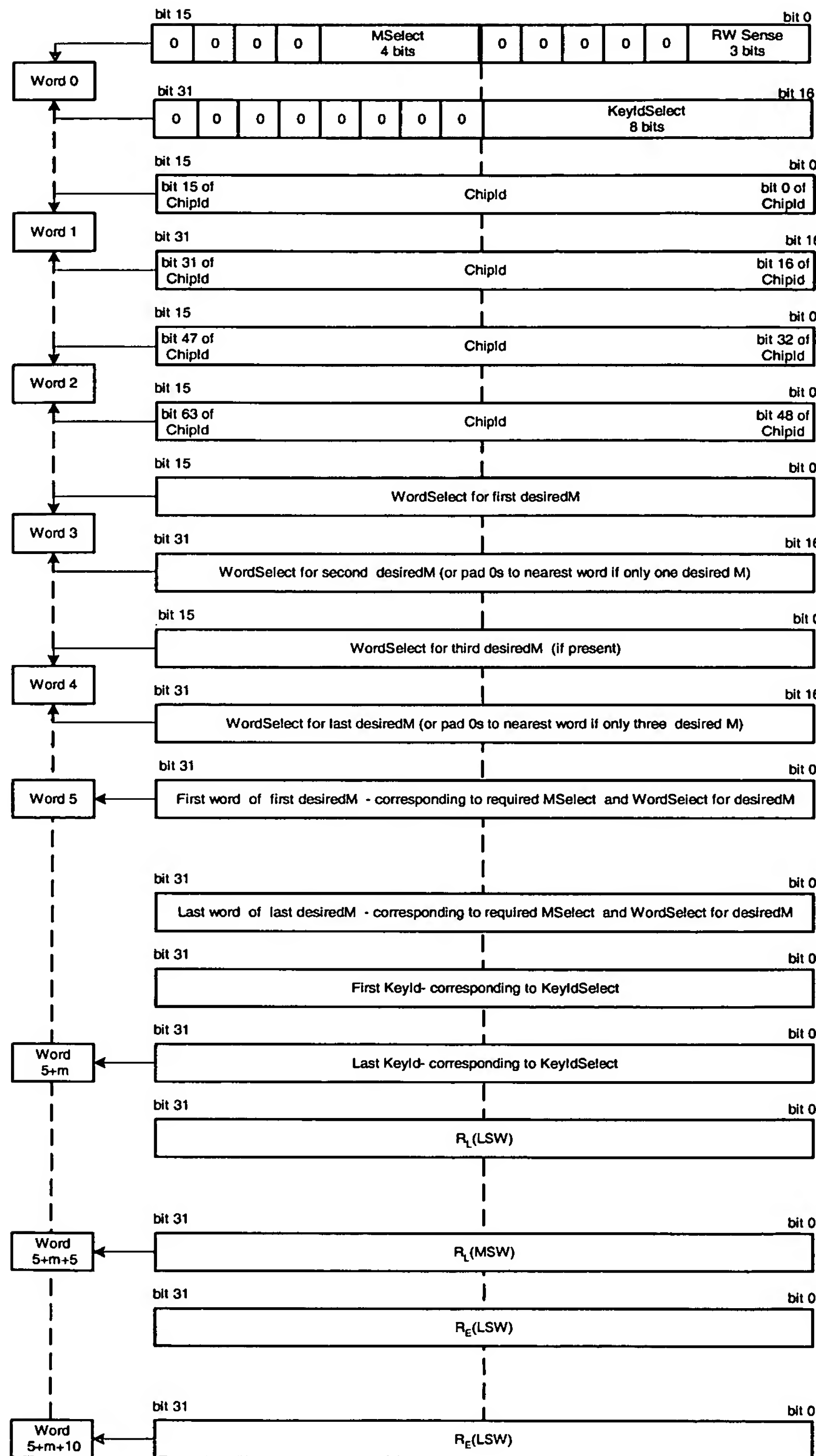


FIG. 370

301/331

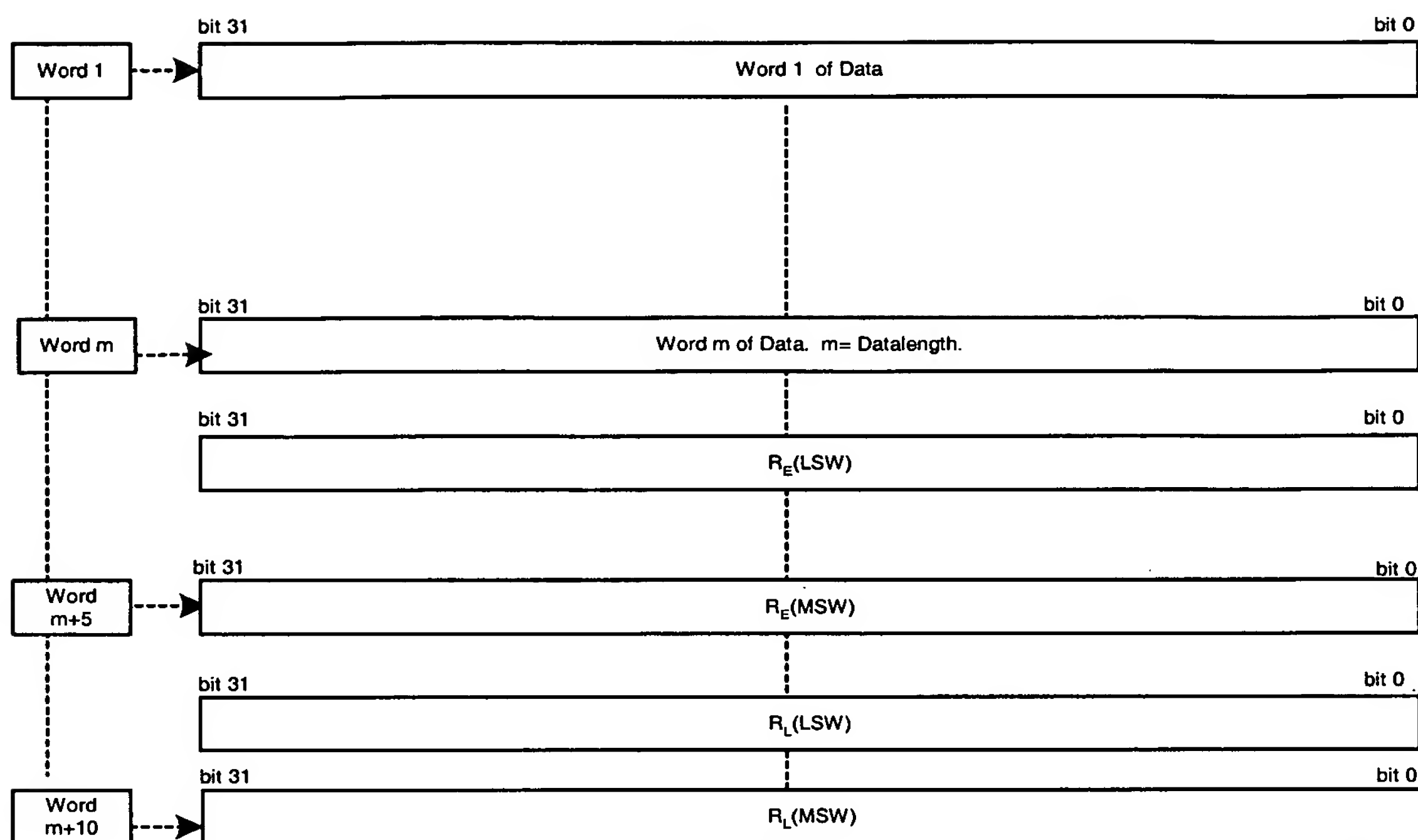


FIG. 371

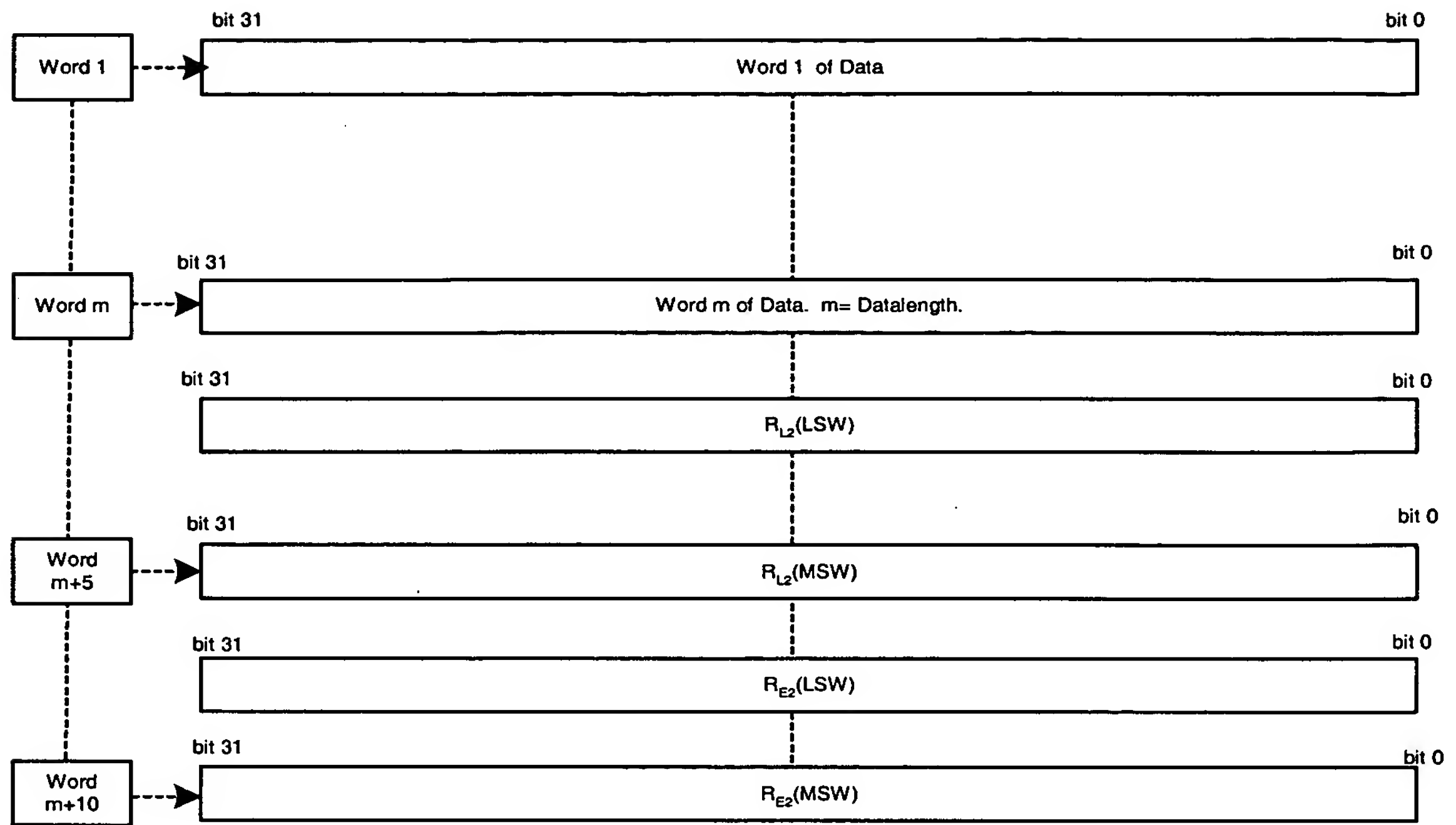


FIG. 372

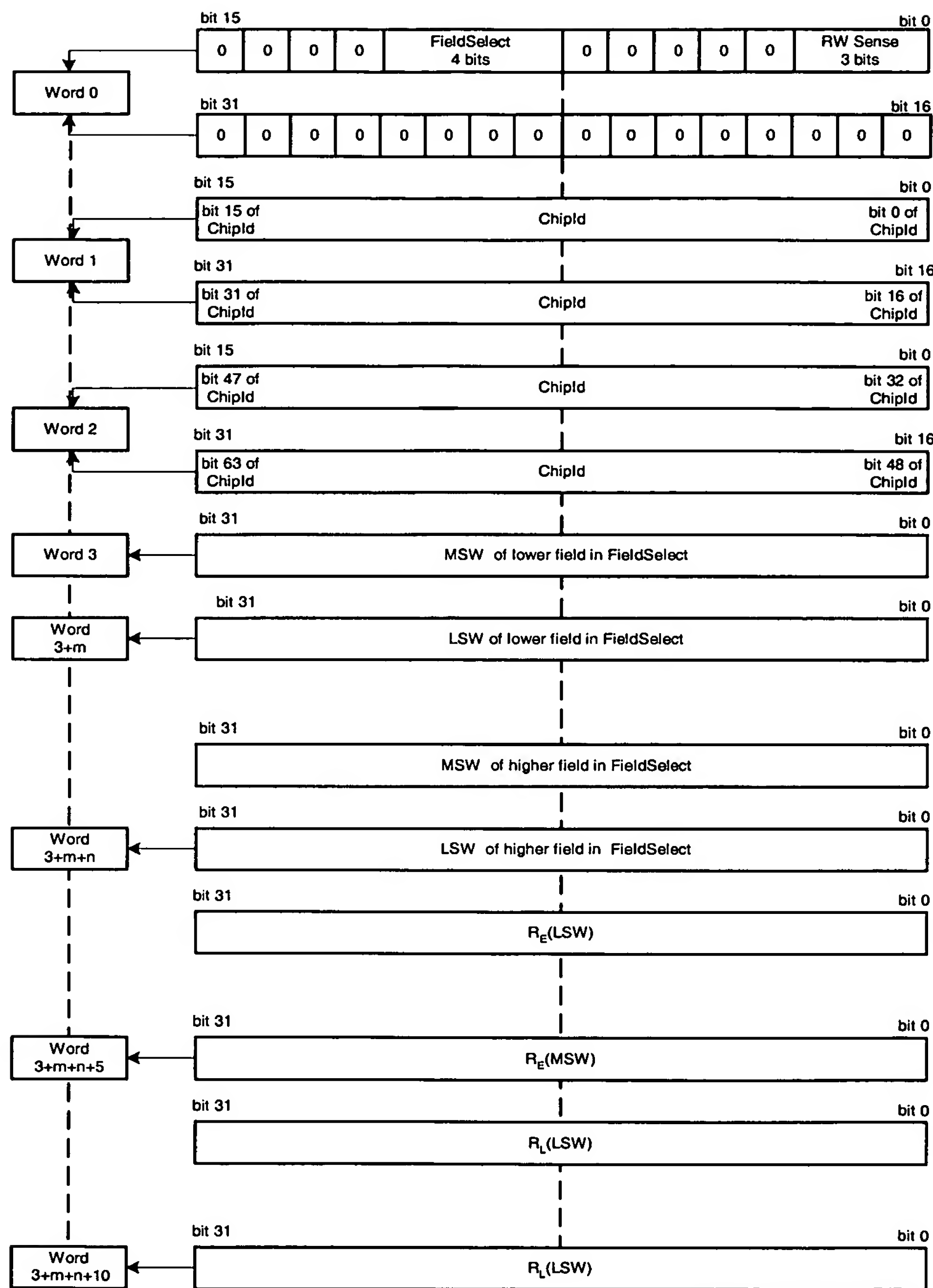


FIG. 373

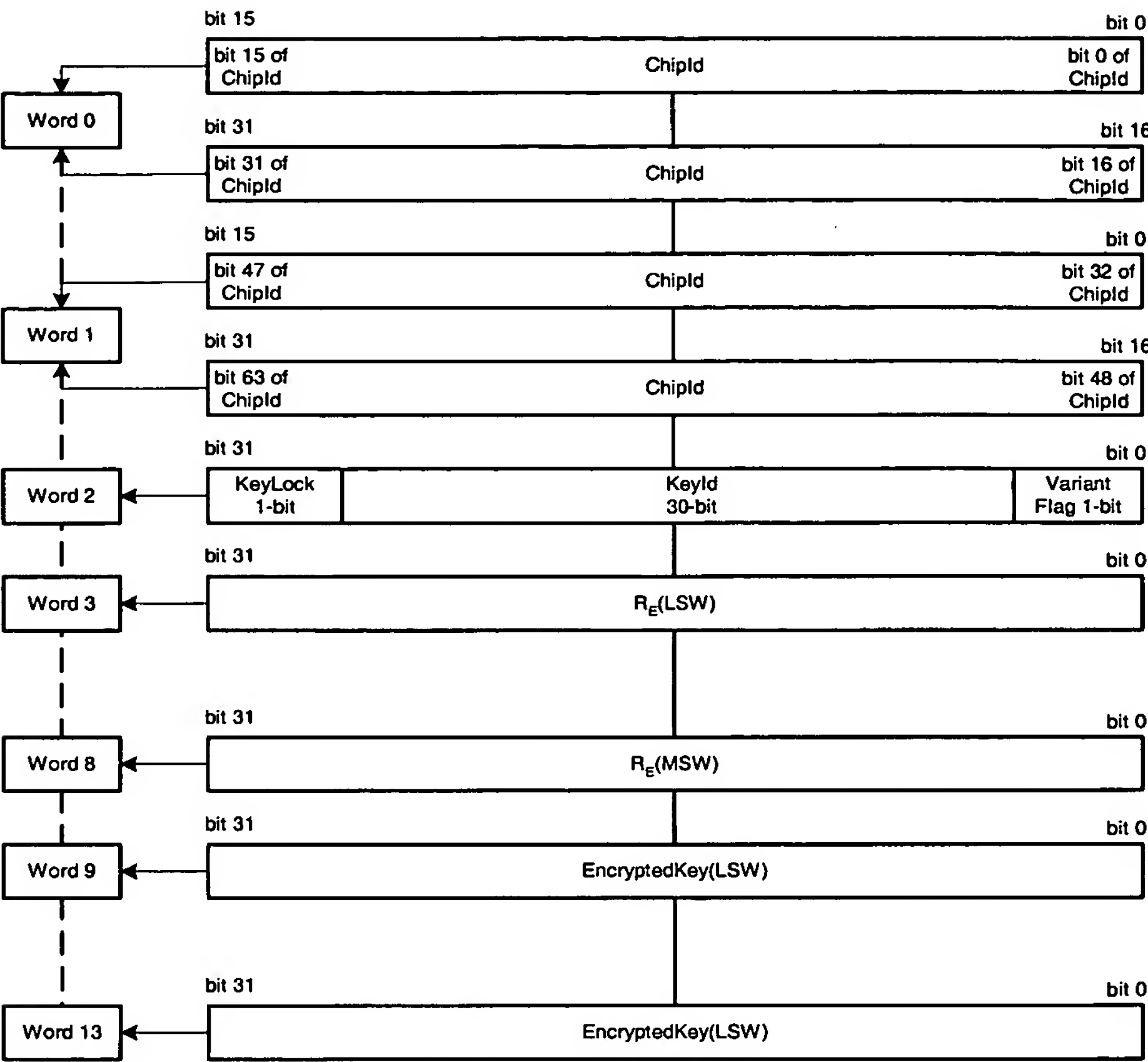


FIG. 374

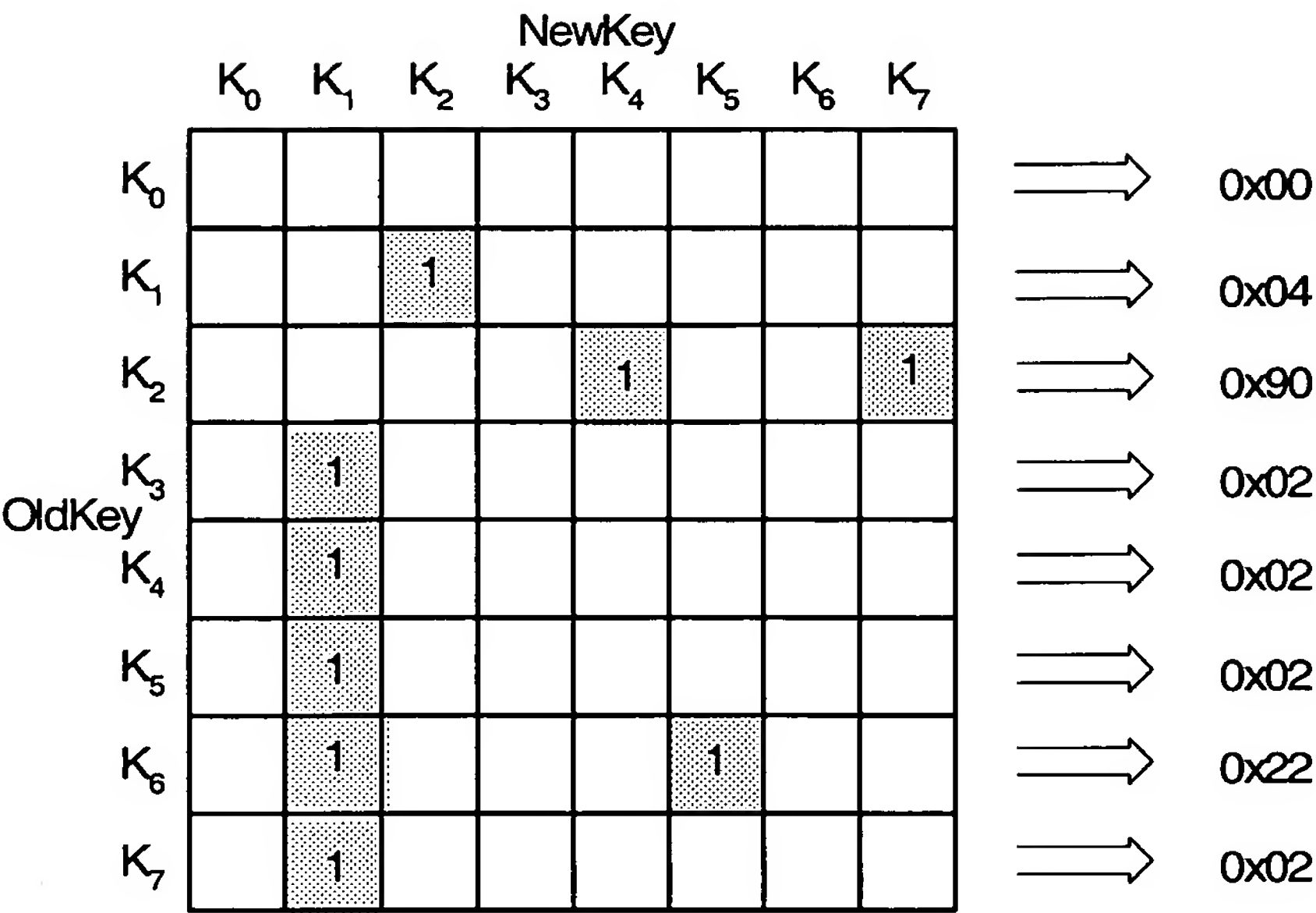


FIG. 375

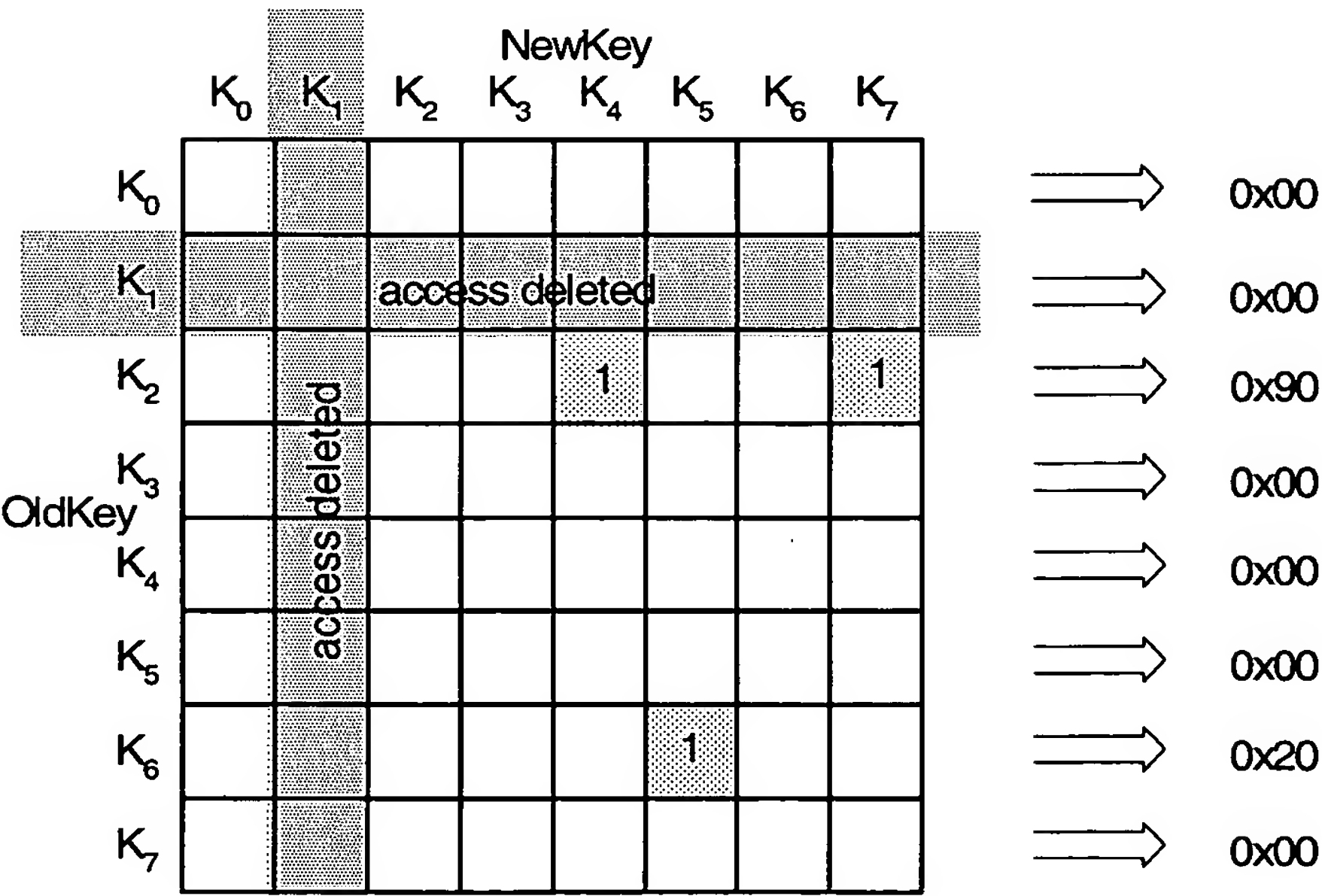


FIG. 376

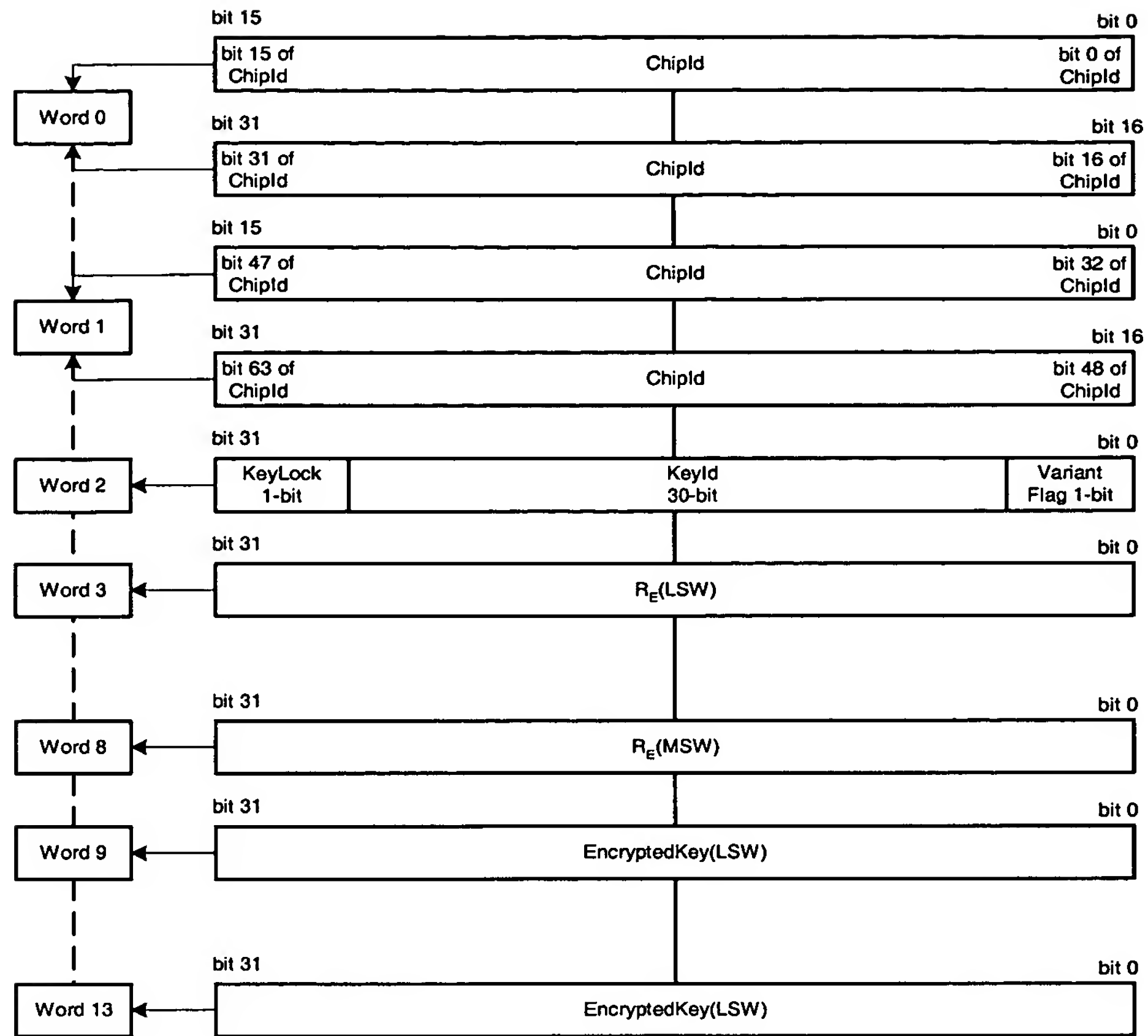


FIG. 377

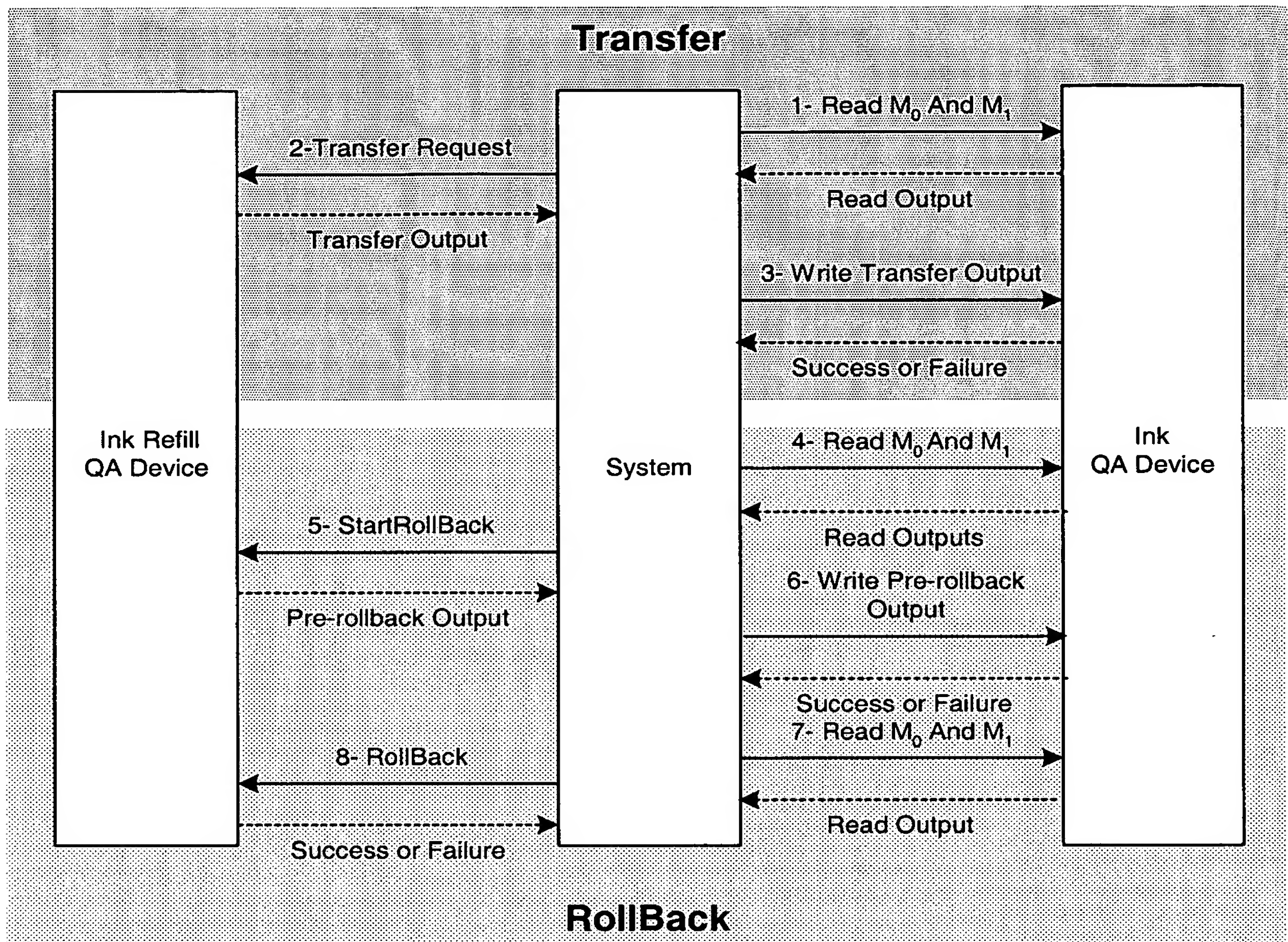


FIG. 378

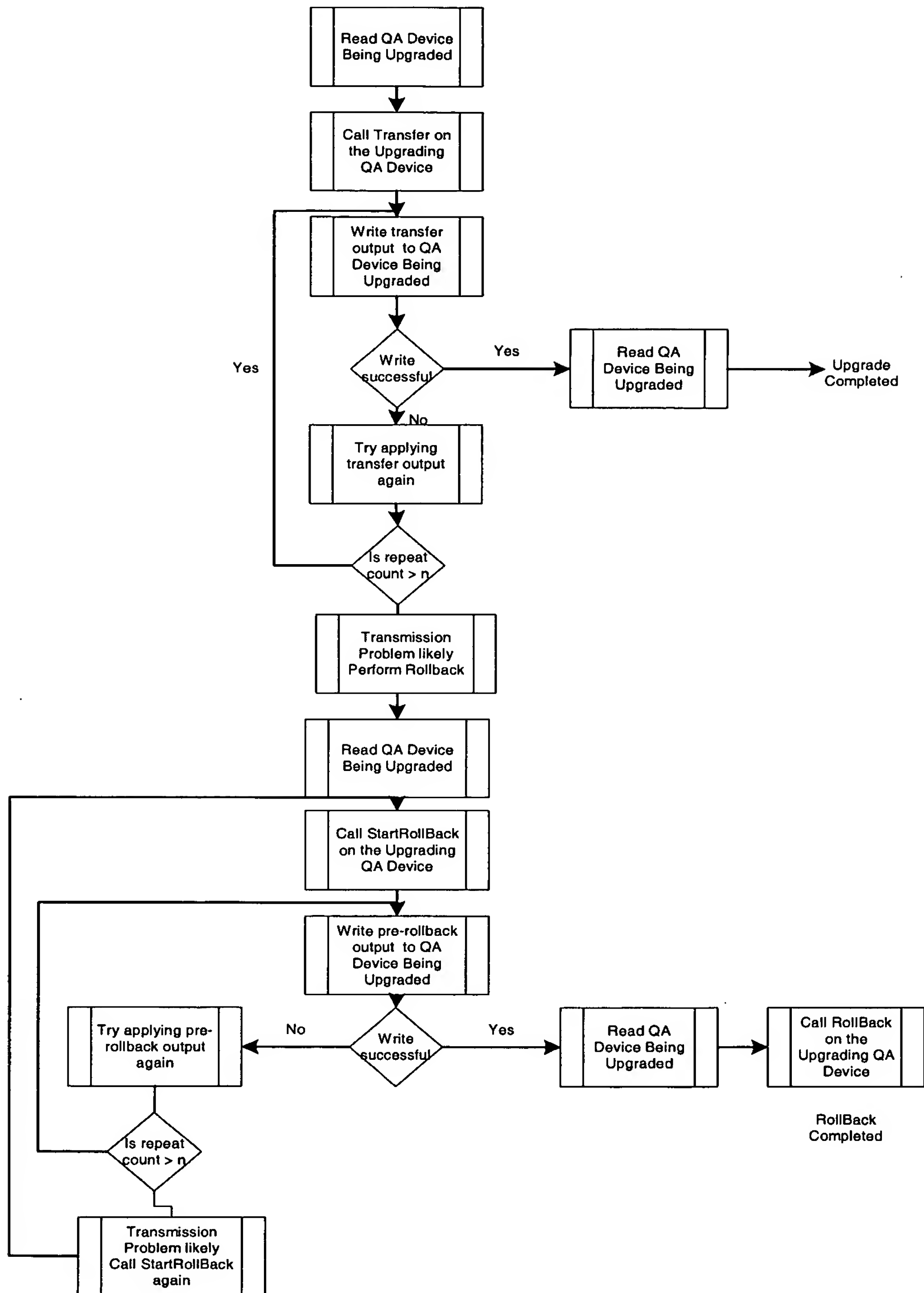


FIG. 379

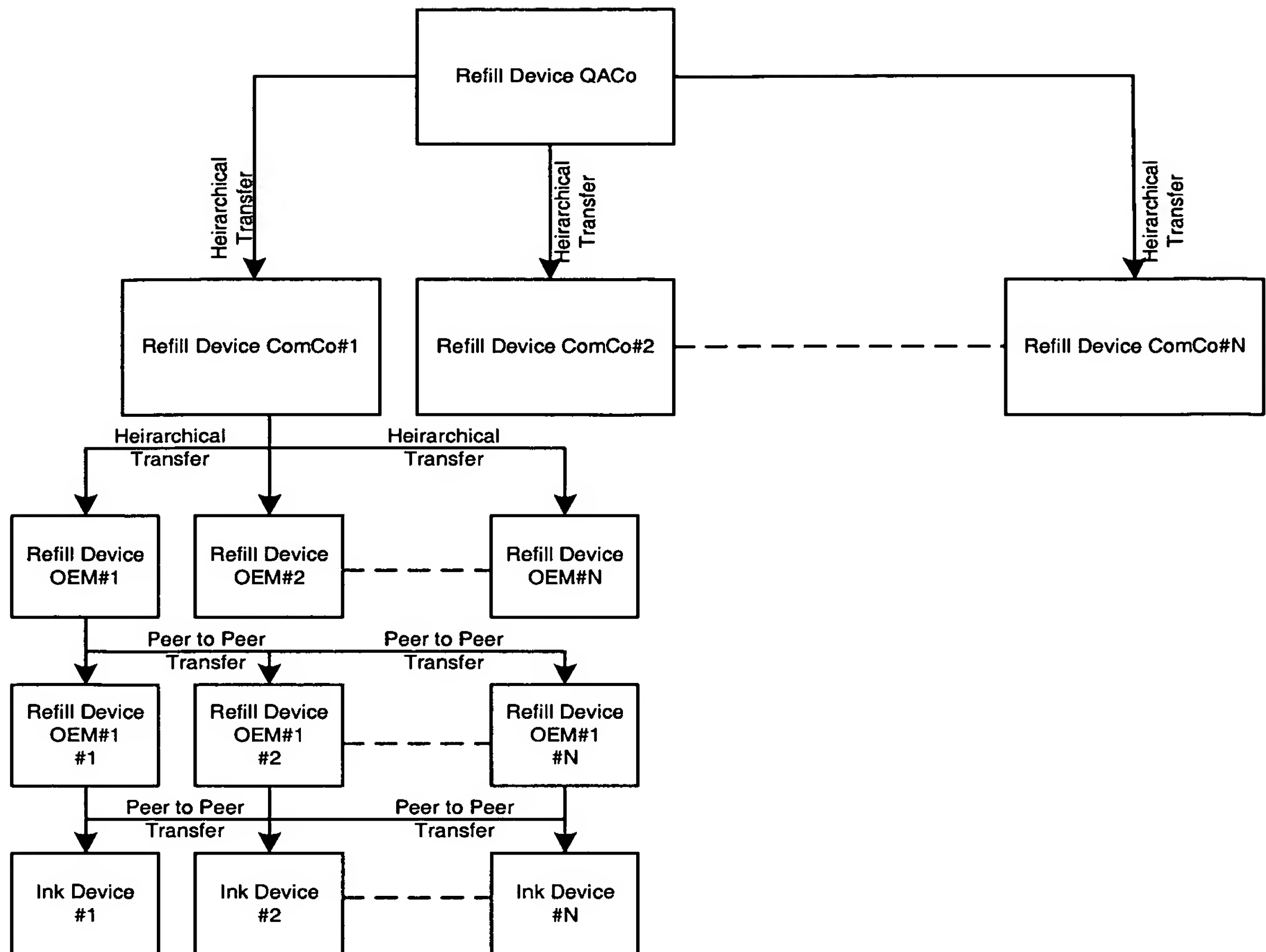


FIG. 380

310/331

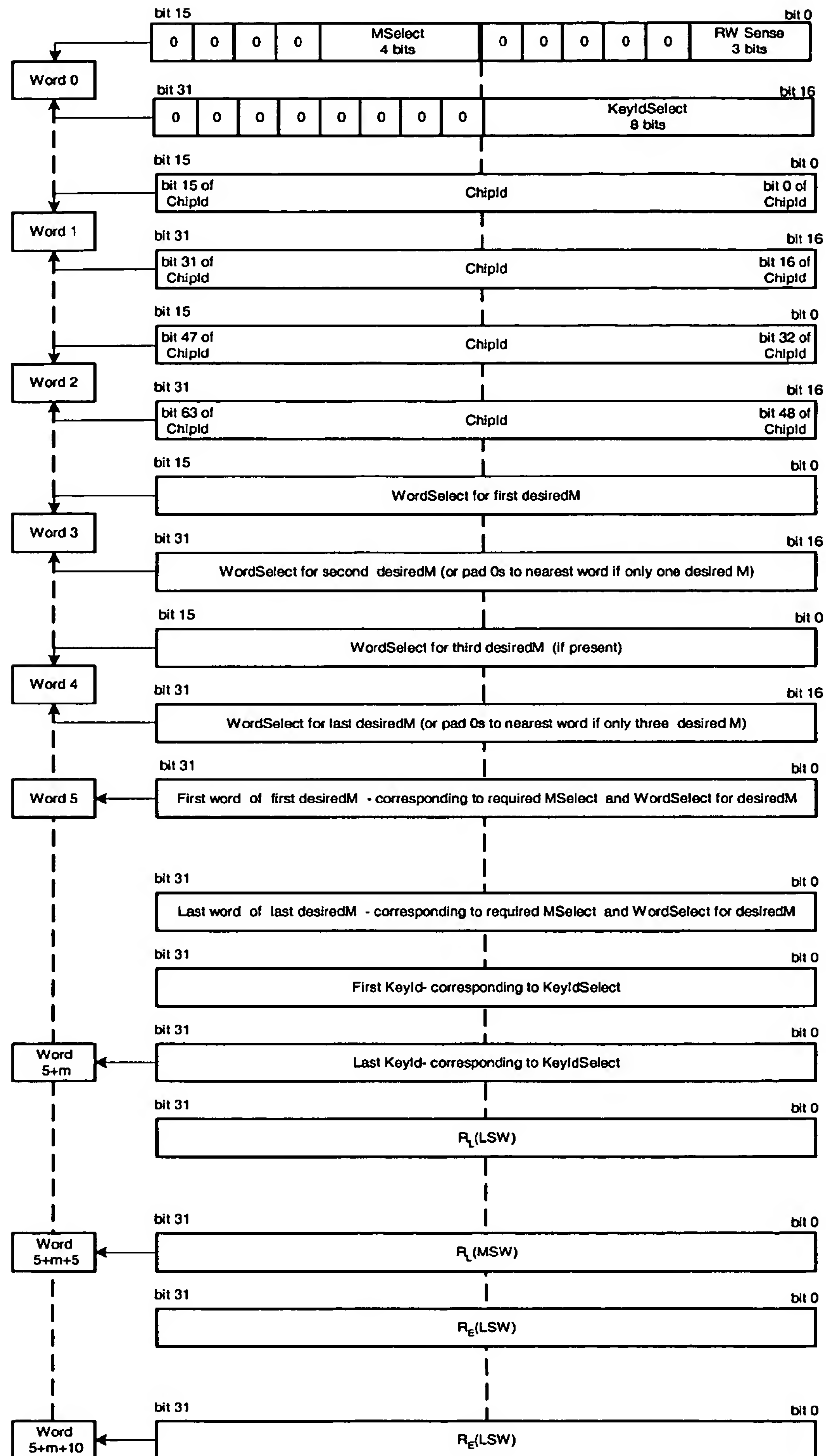


FIG. 381

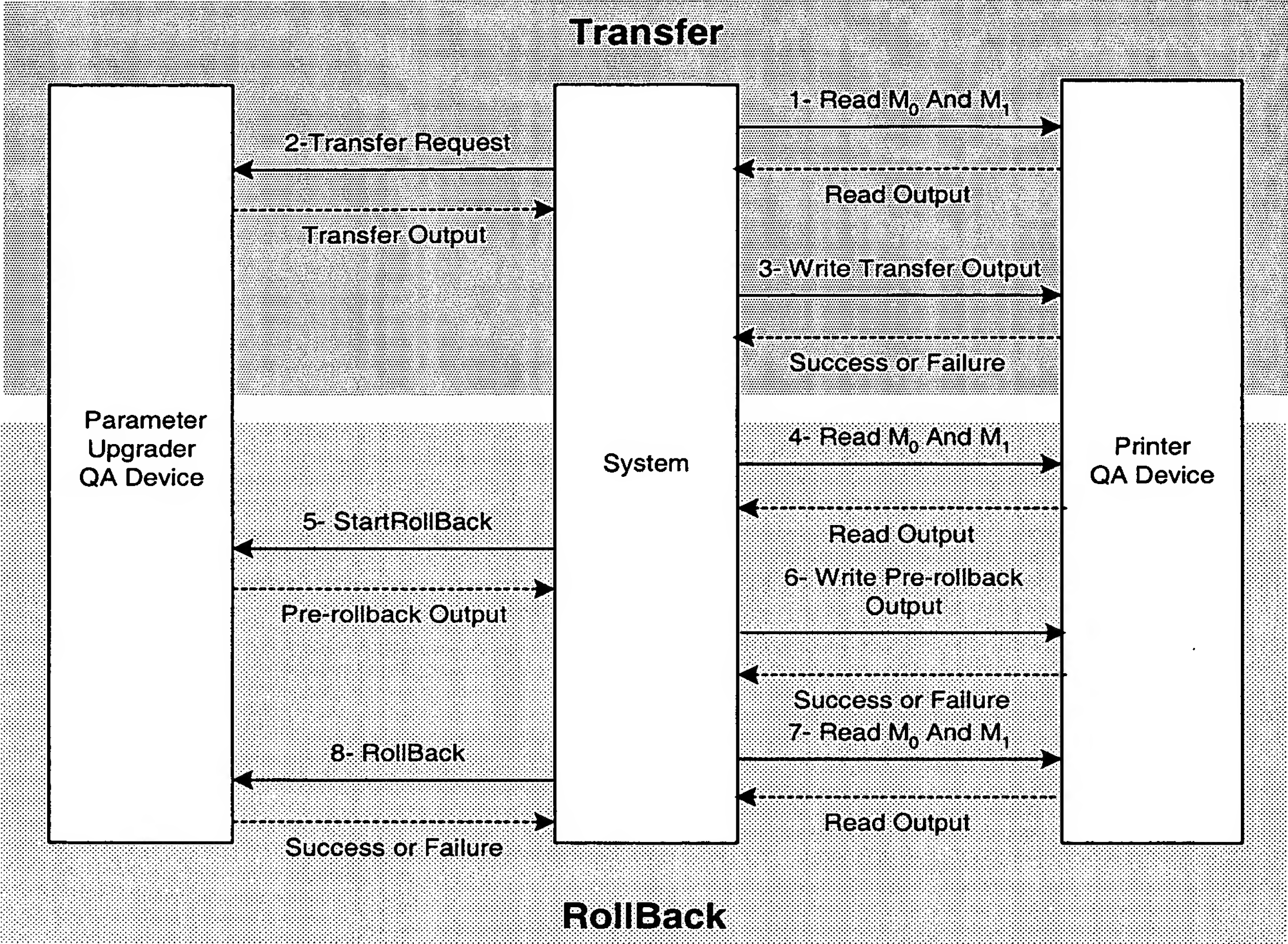


FIG. 382

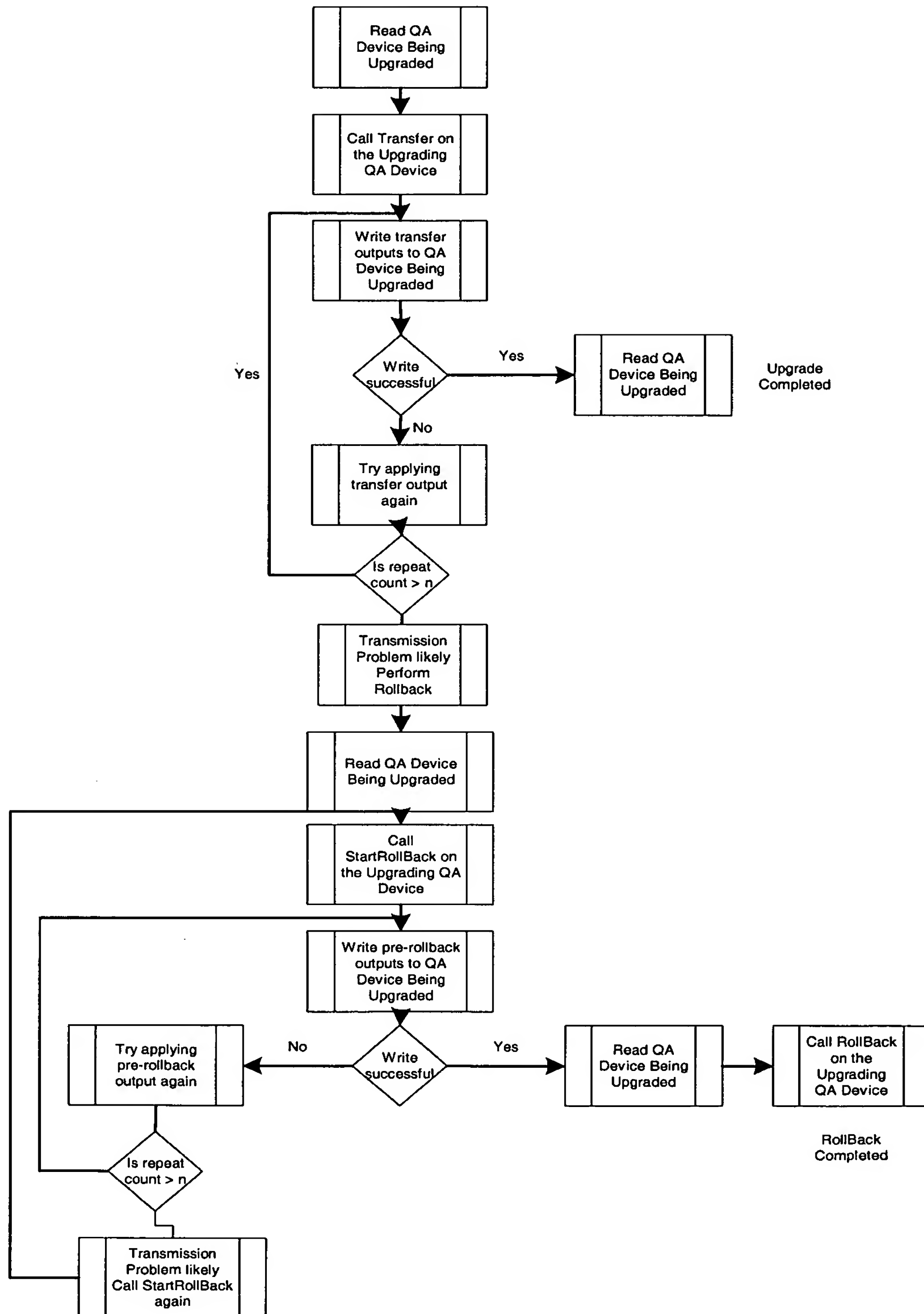


FIG. 383

313/331

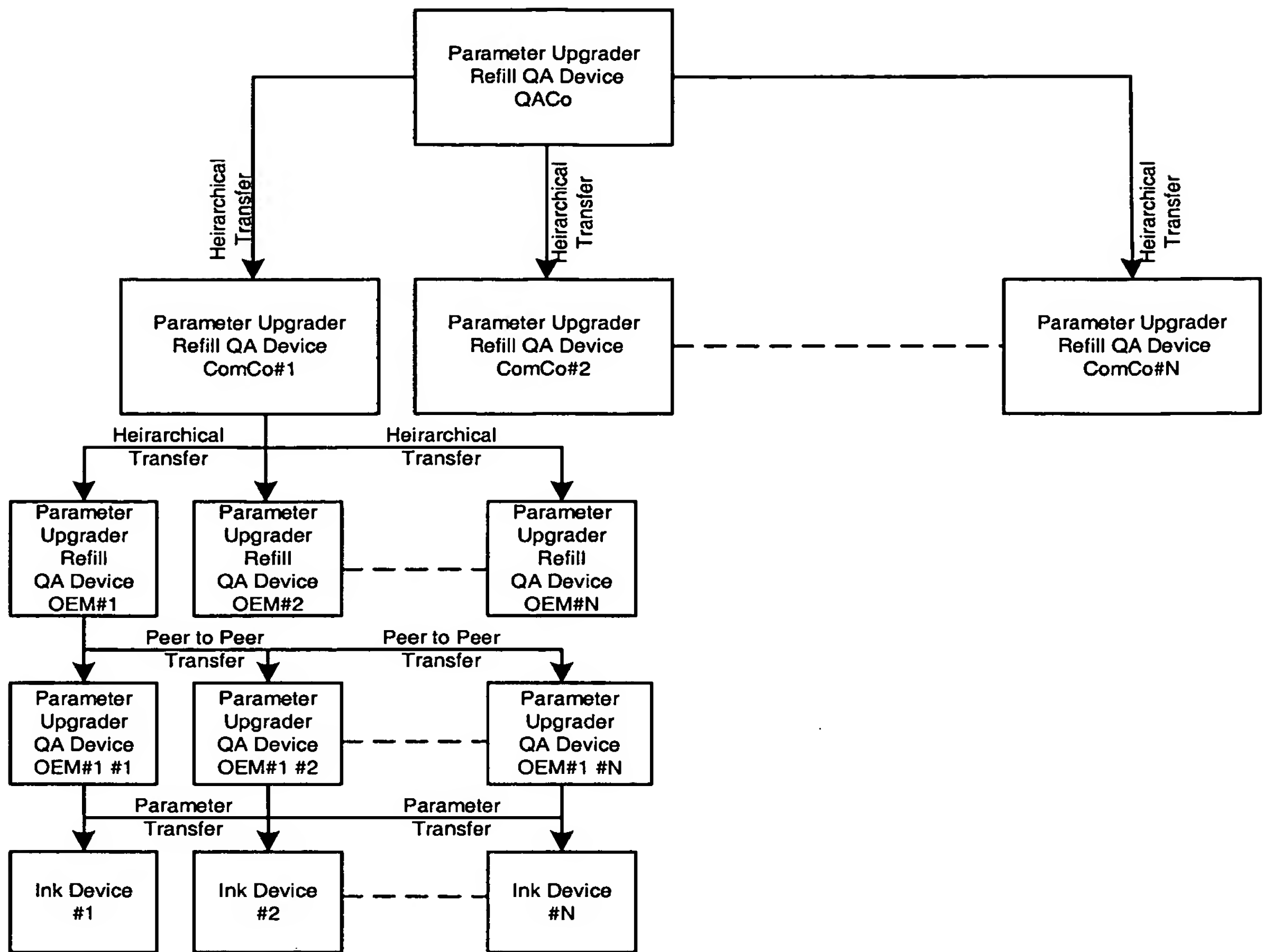


FIG. 384

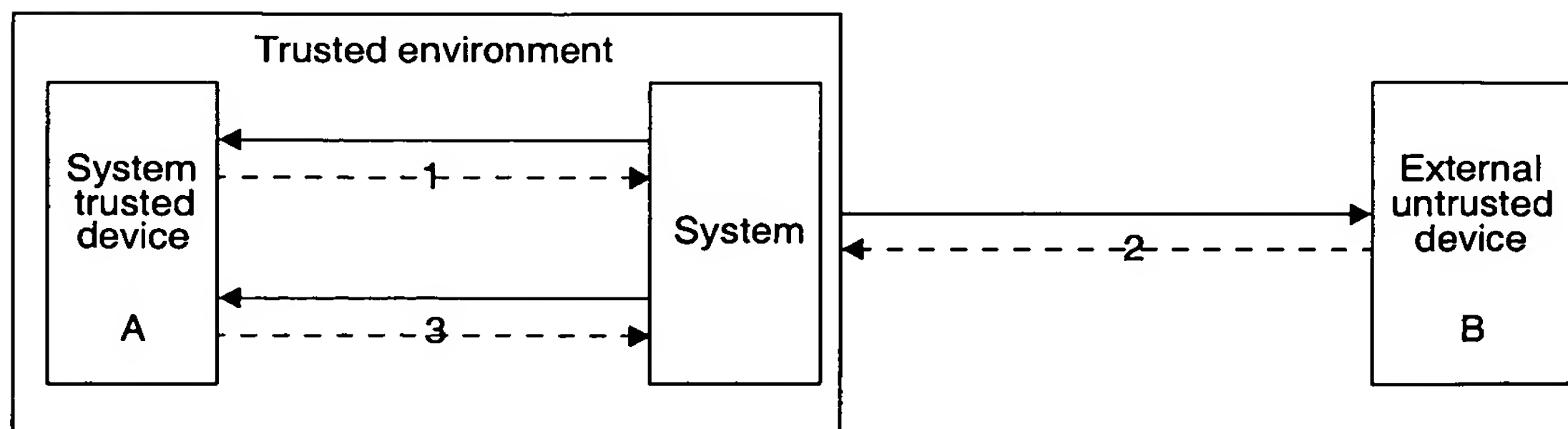
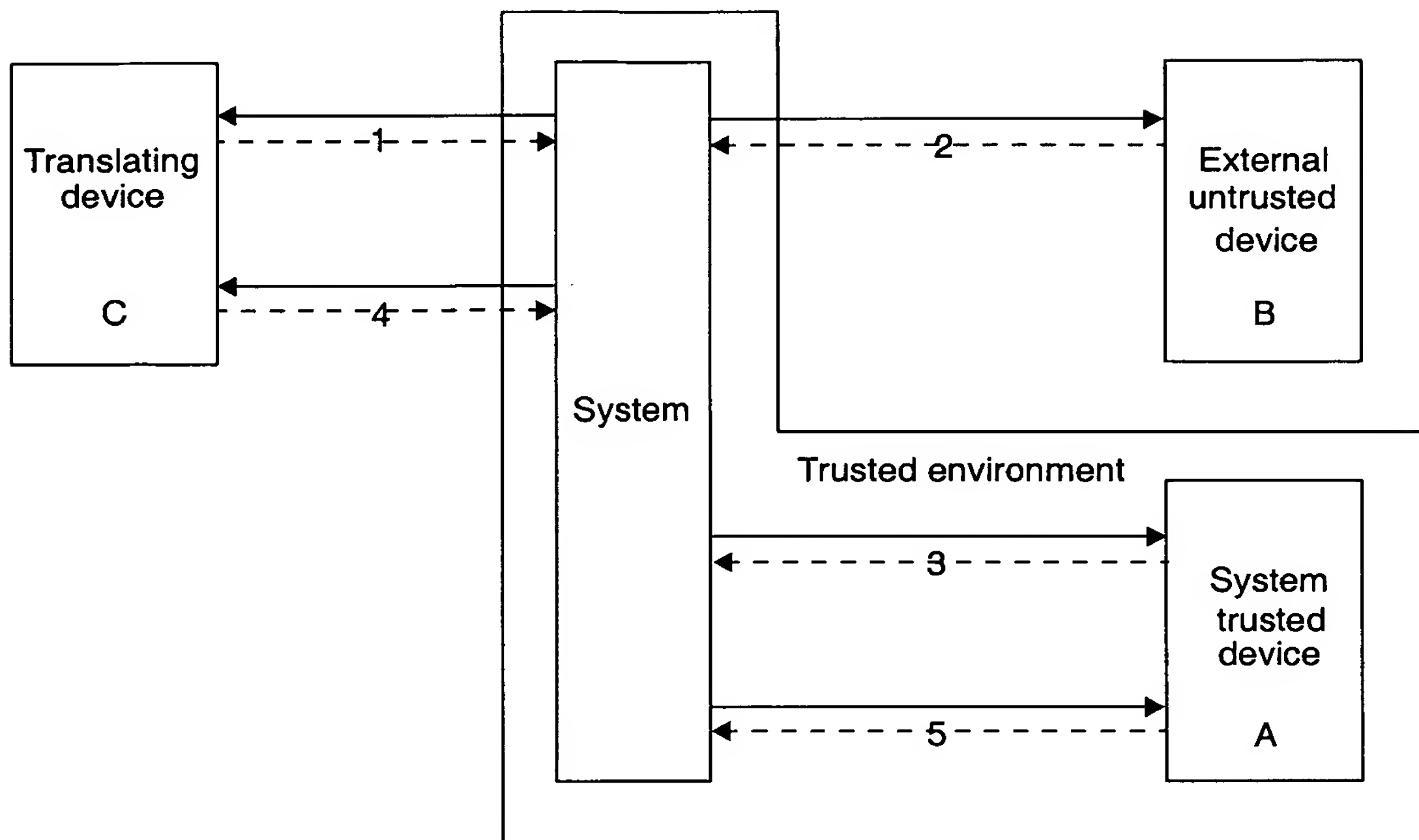


FIG. 385

*FIG. 386*

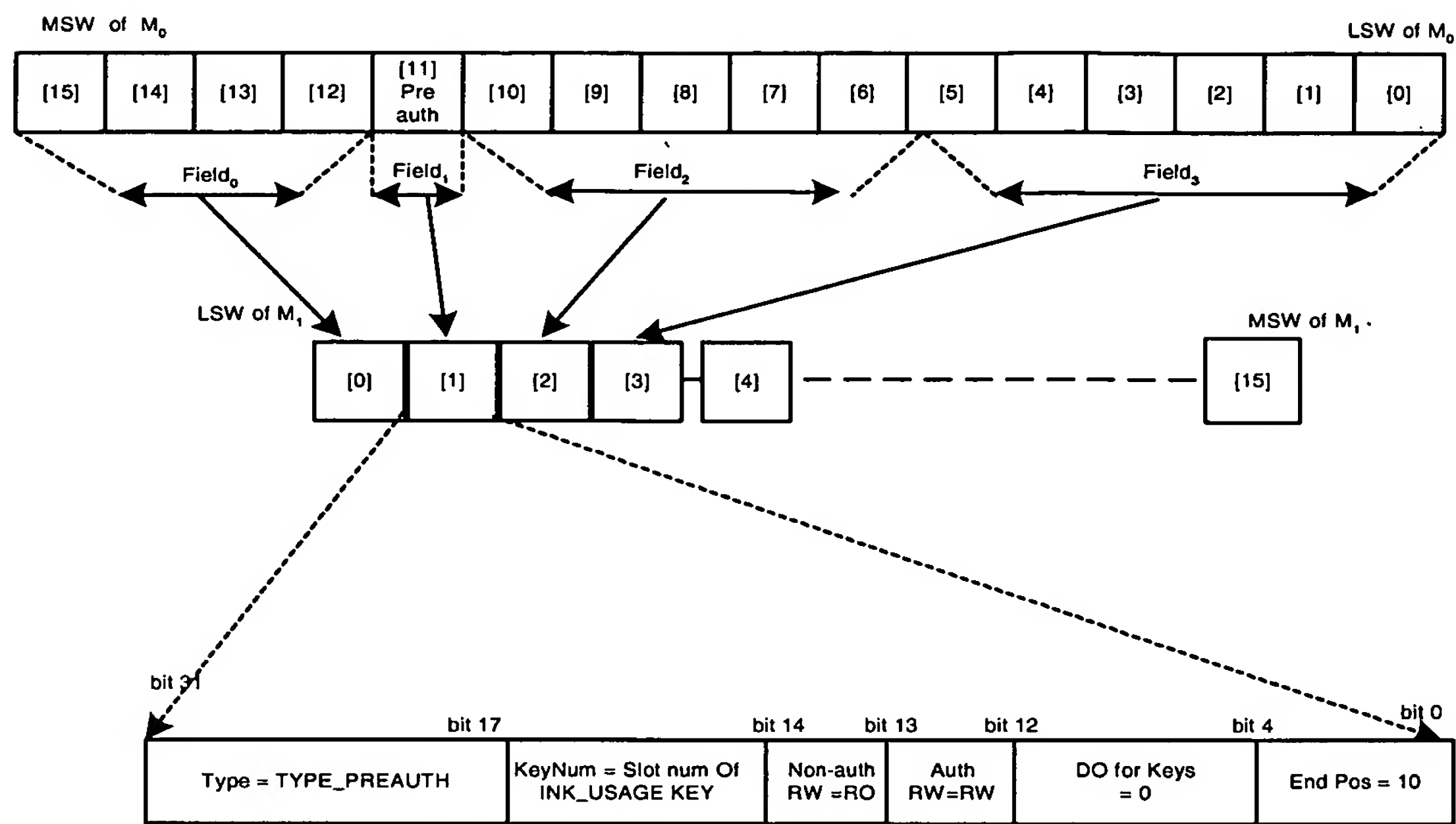


FIG. 387

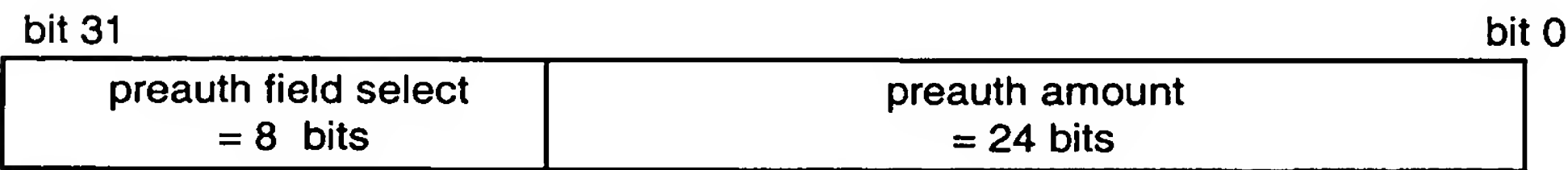


FIG. 388

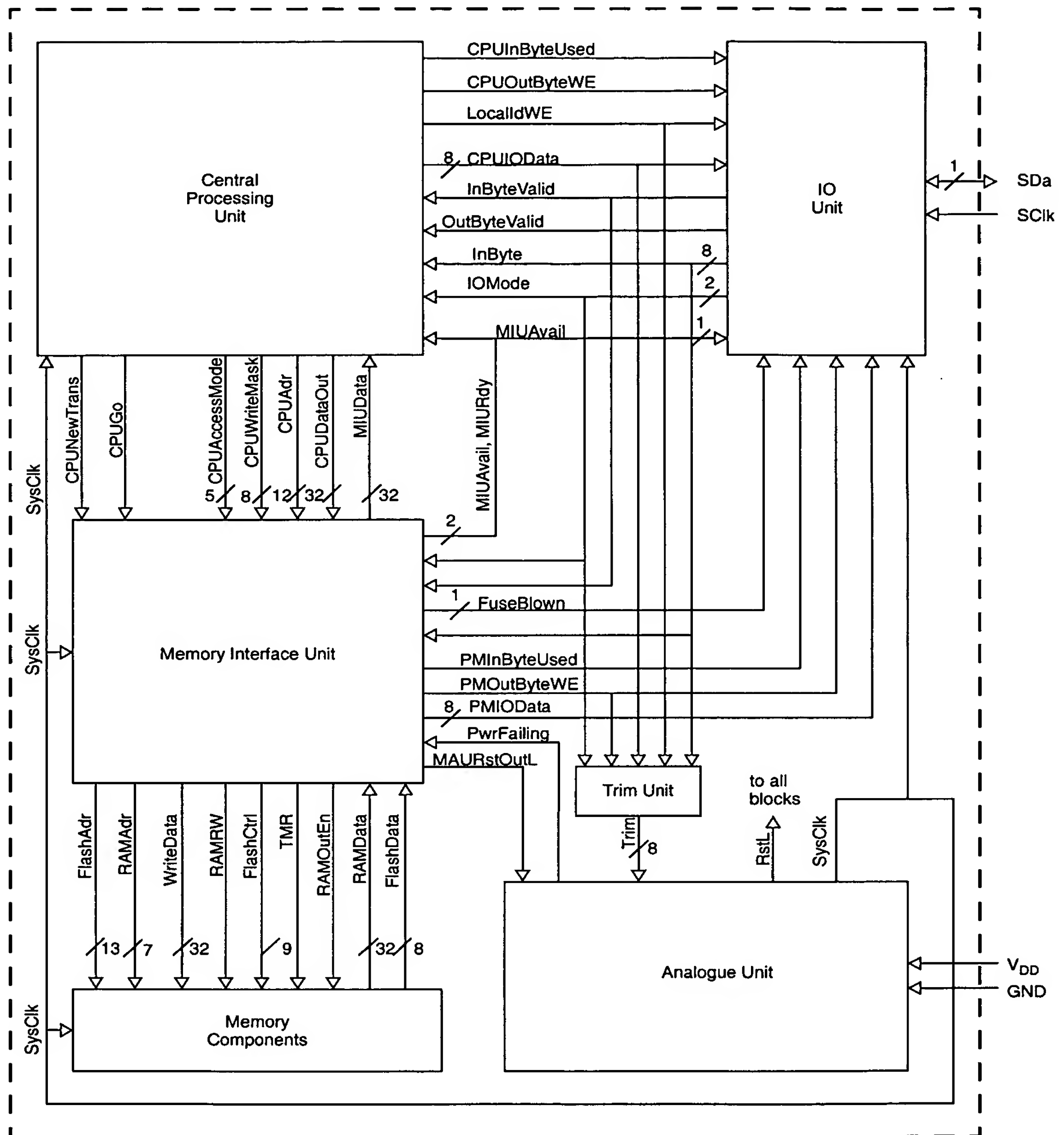


FIG. 389

317/331

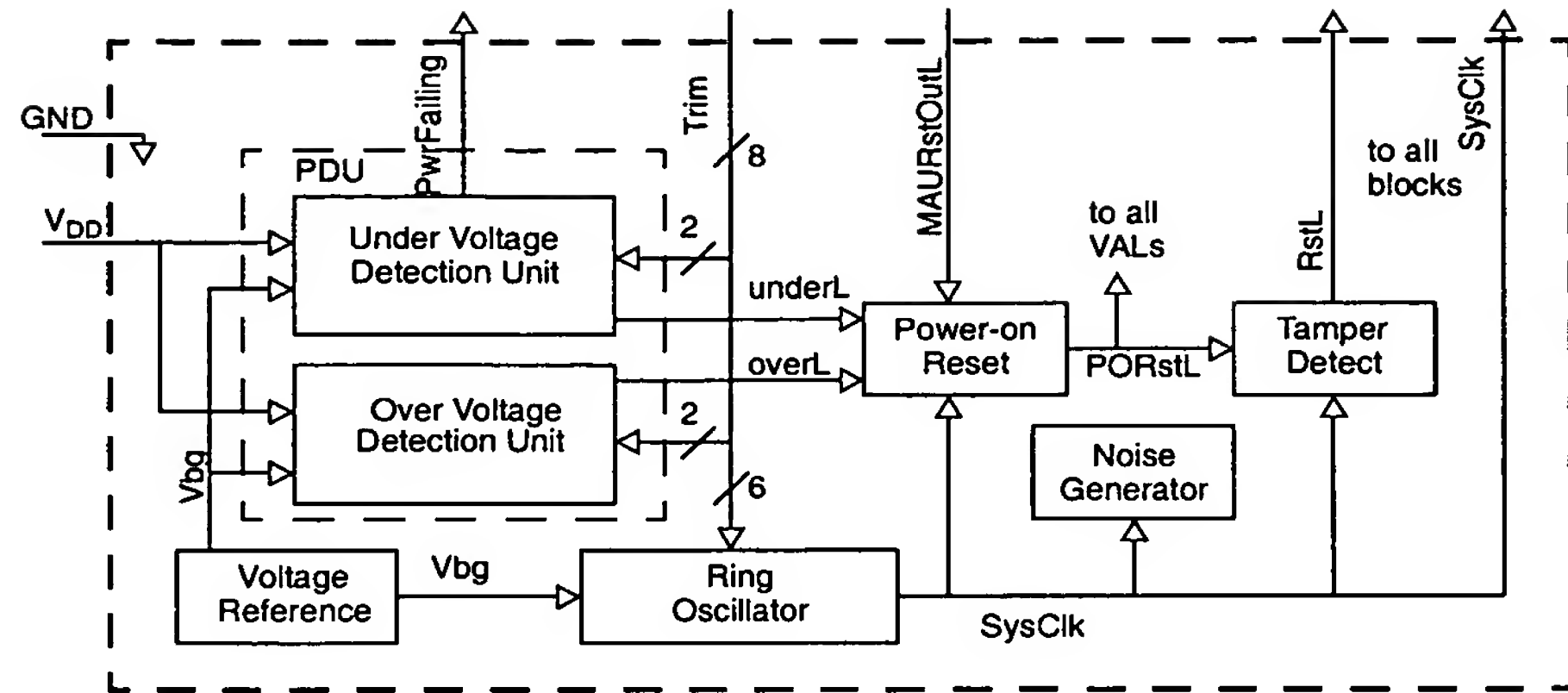


FIG. 390

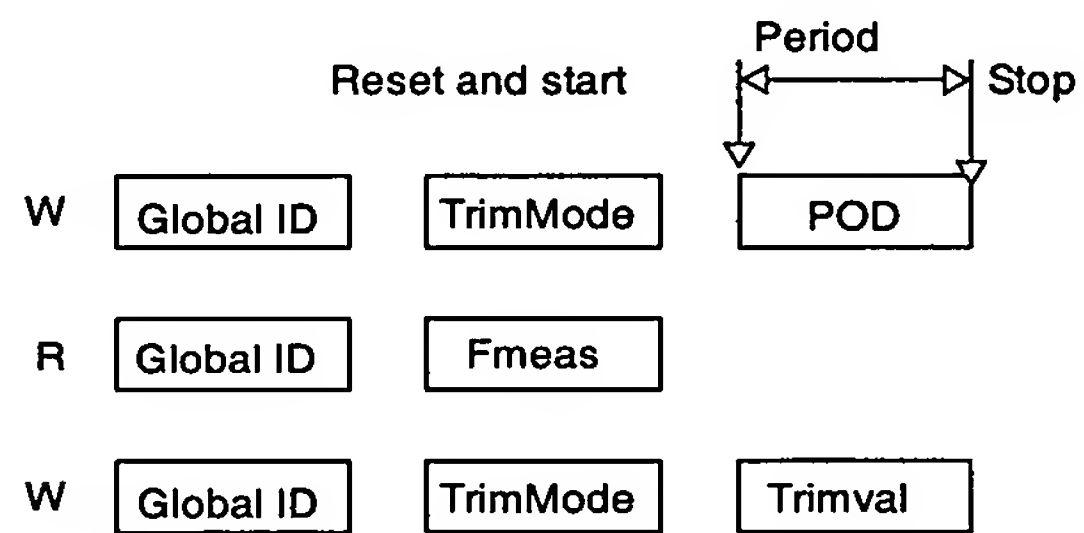


FIG. 391

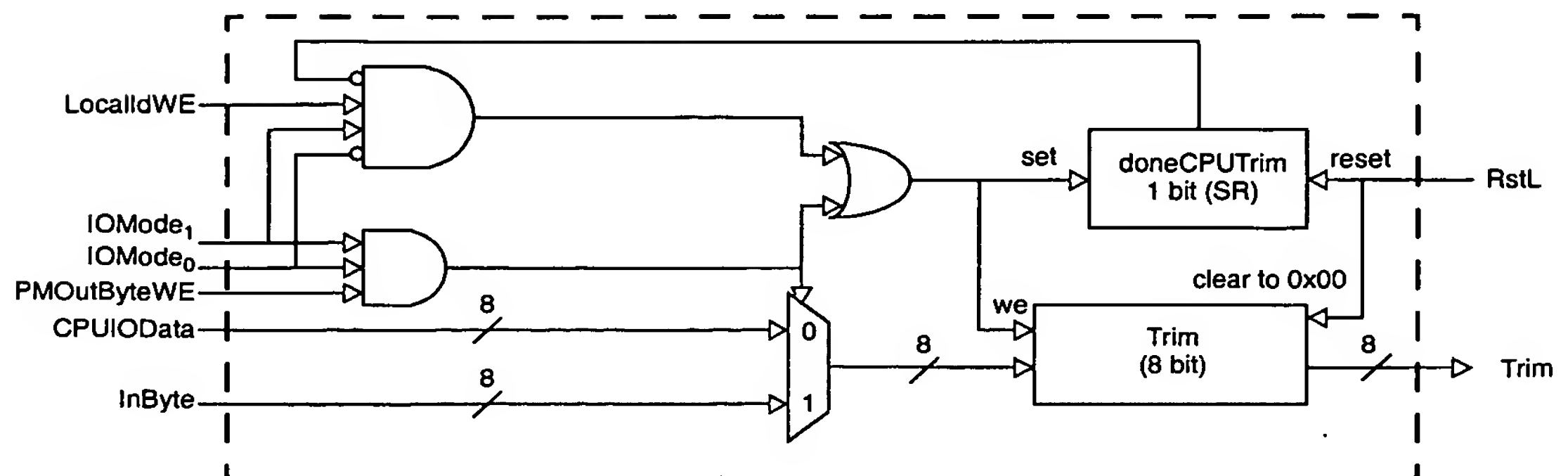


FIG. 392

318/331

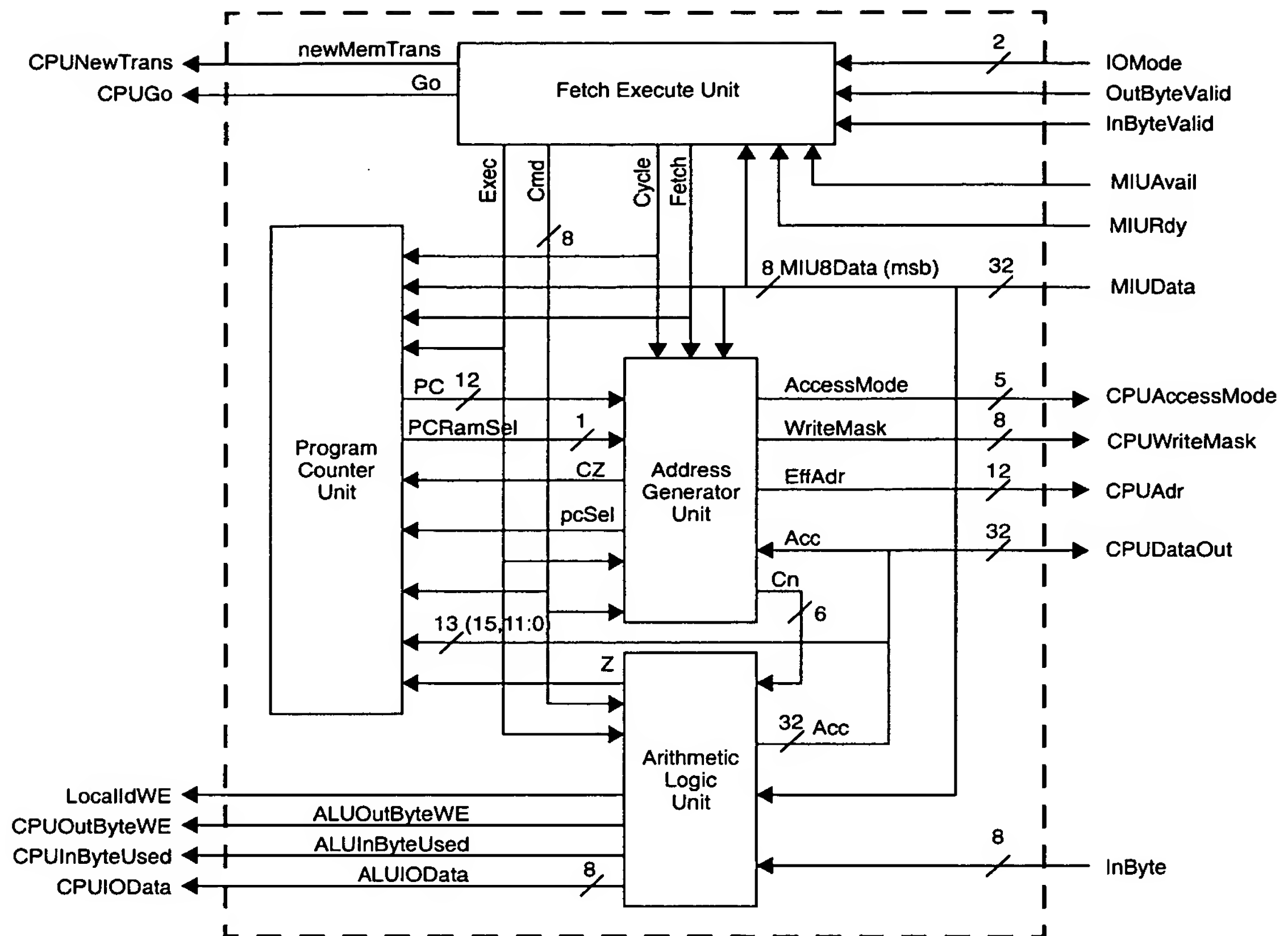


FIG. 393

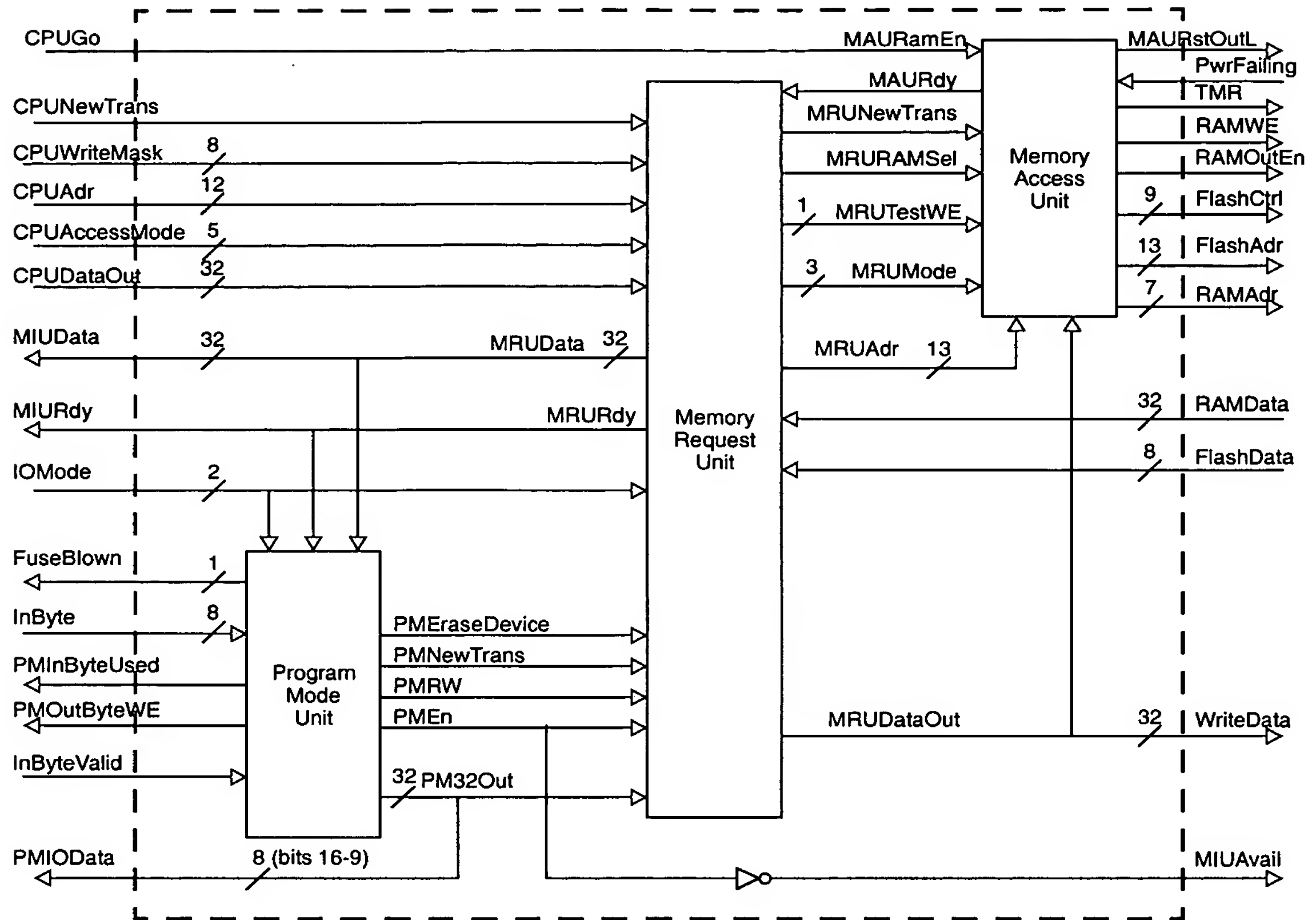


FIG. 394

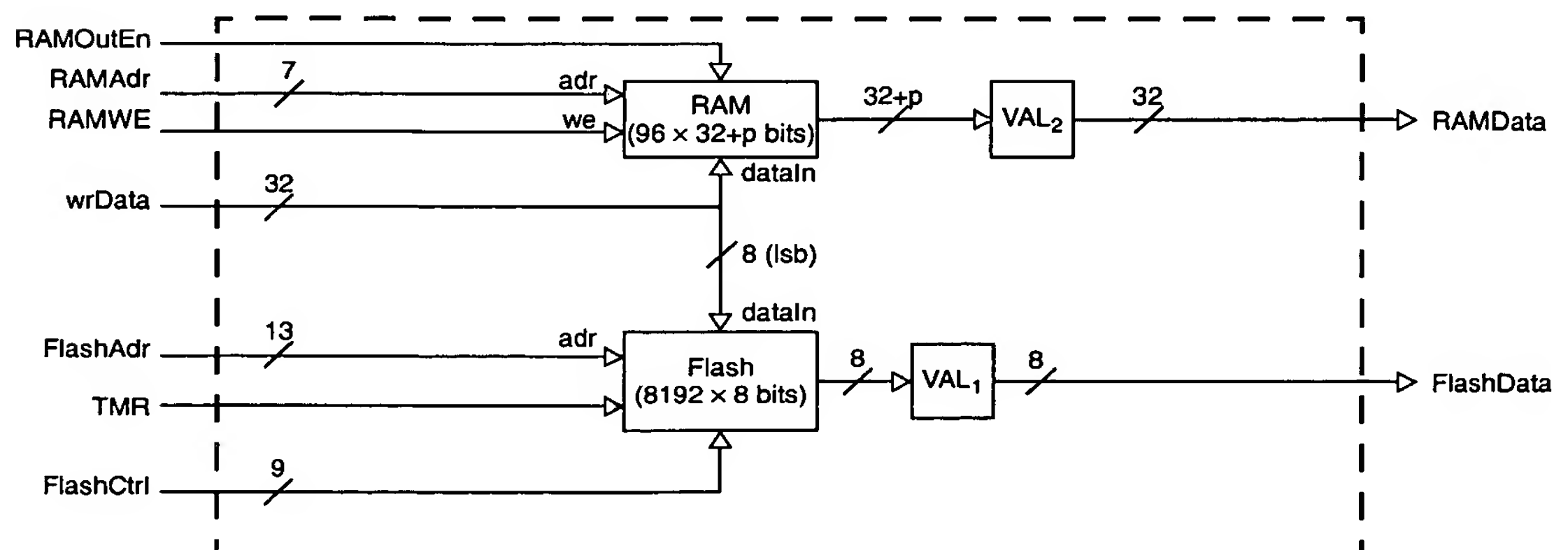


FIG. 395

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PrID6	PrID5	PrID4	PrID3	PrID2	PrID1	PrID0	R/*W 0 = write 1 = read

FIG. 396

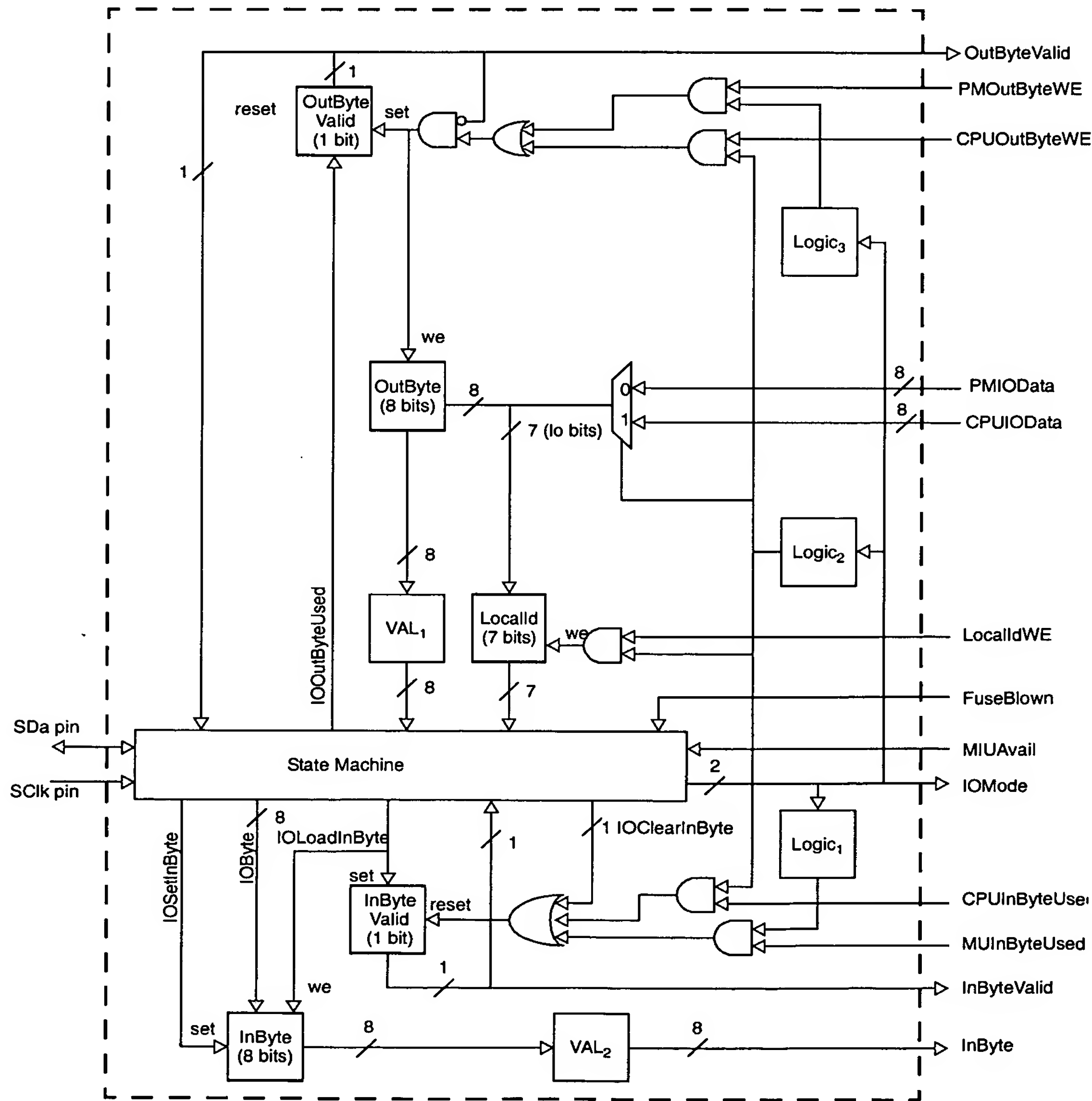


FIG. 397

321/331

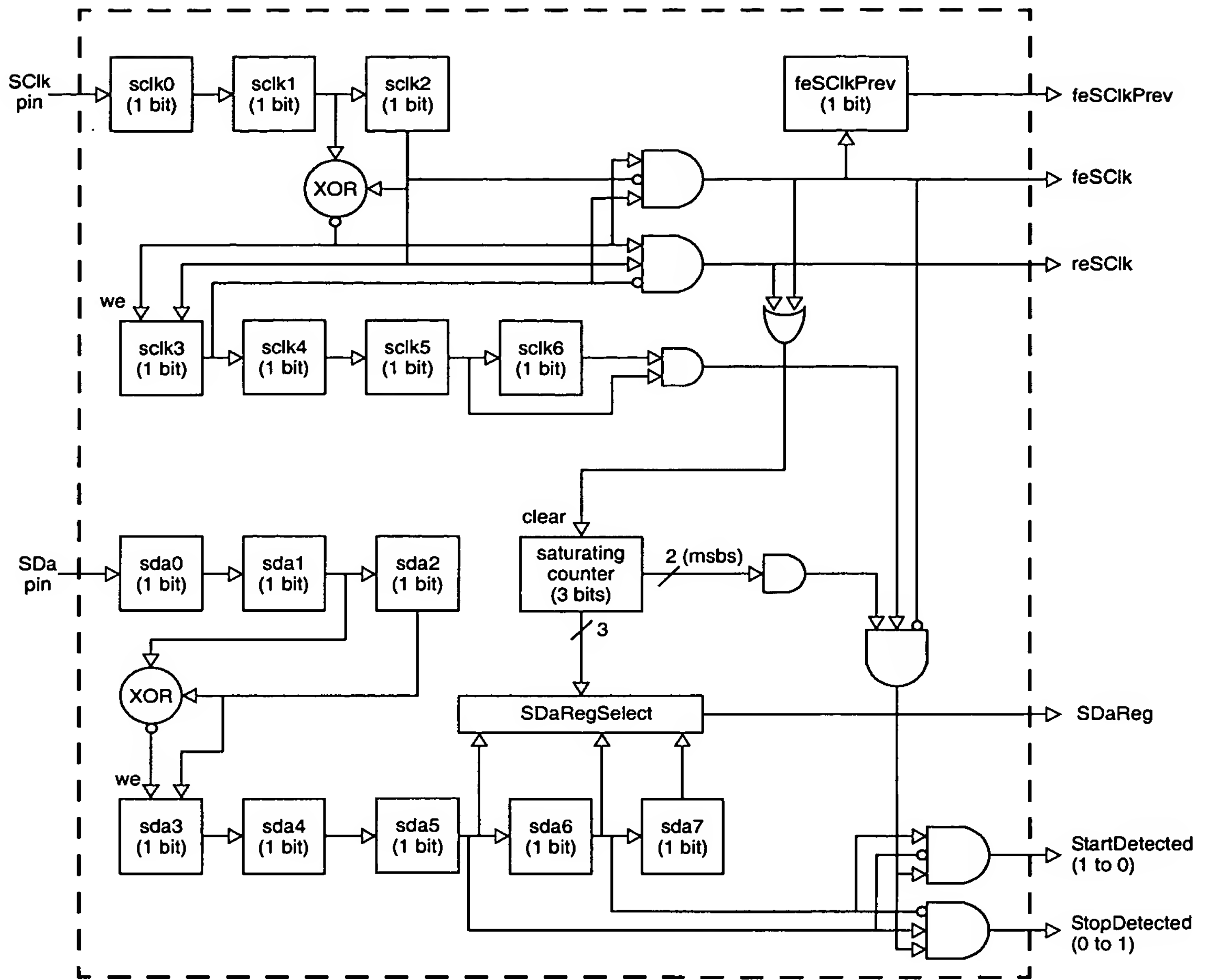


FIG. 398

322/331

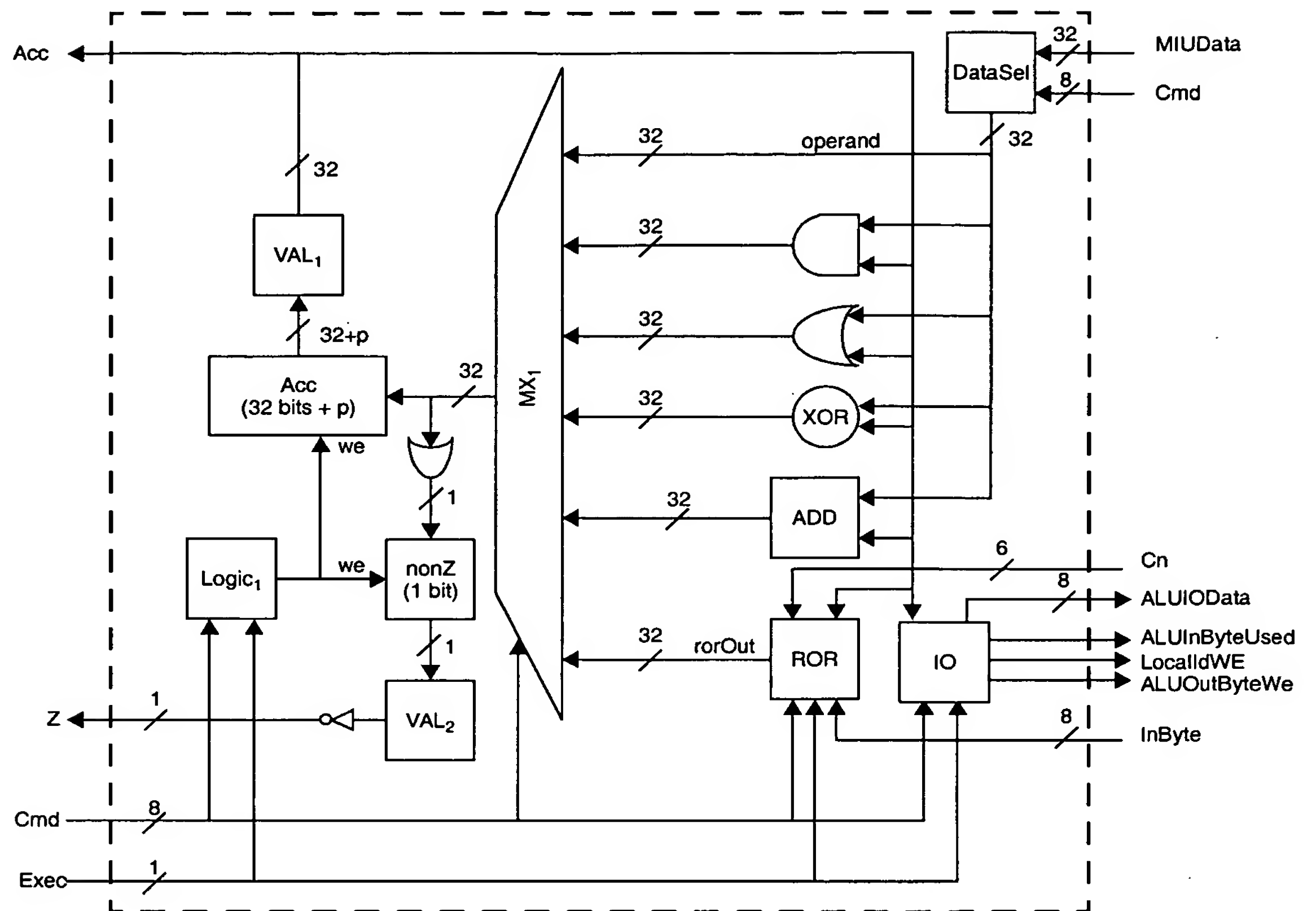


FIG. 399

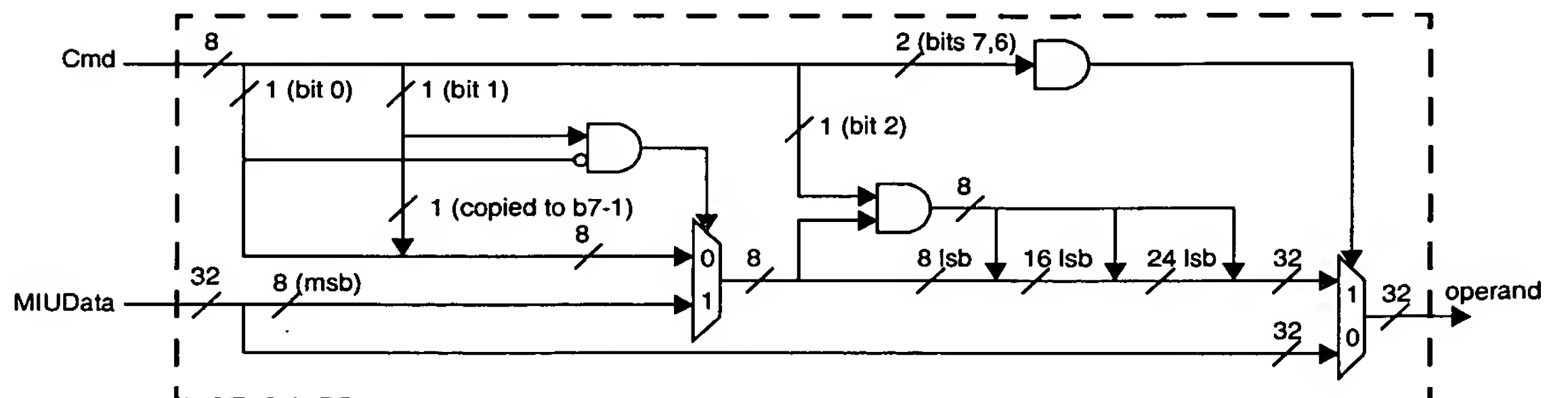


FIG. 400

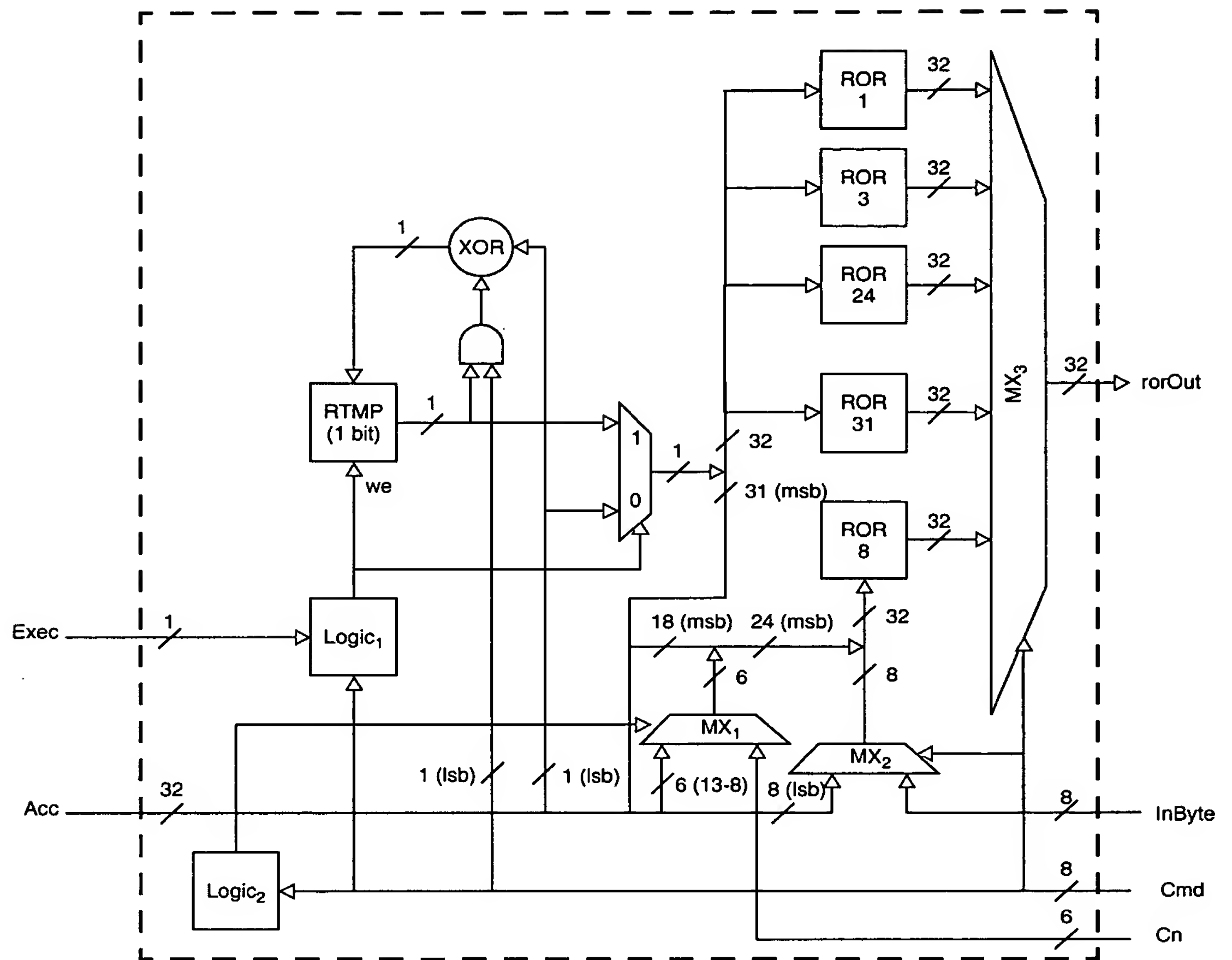


FIG. 401

324/331

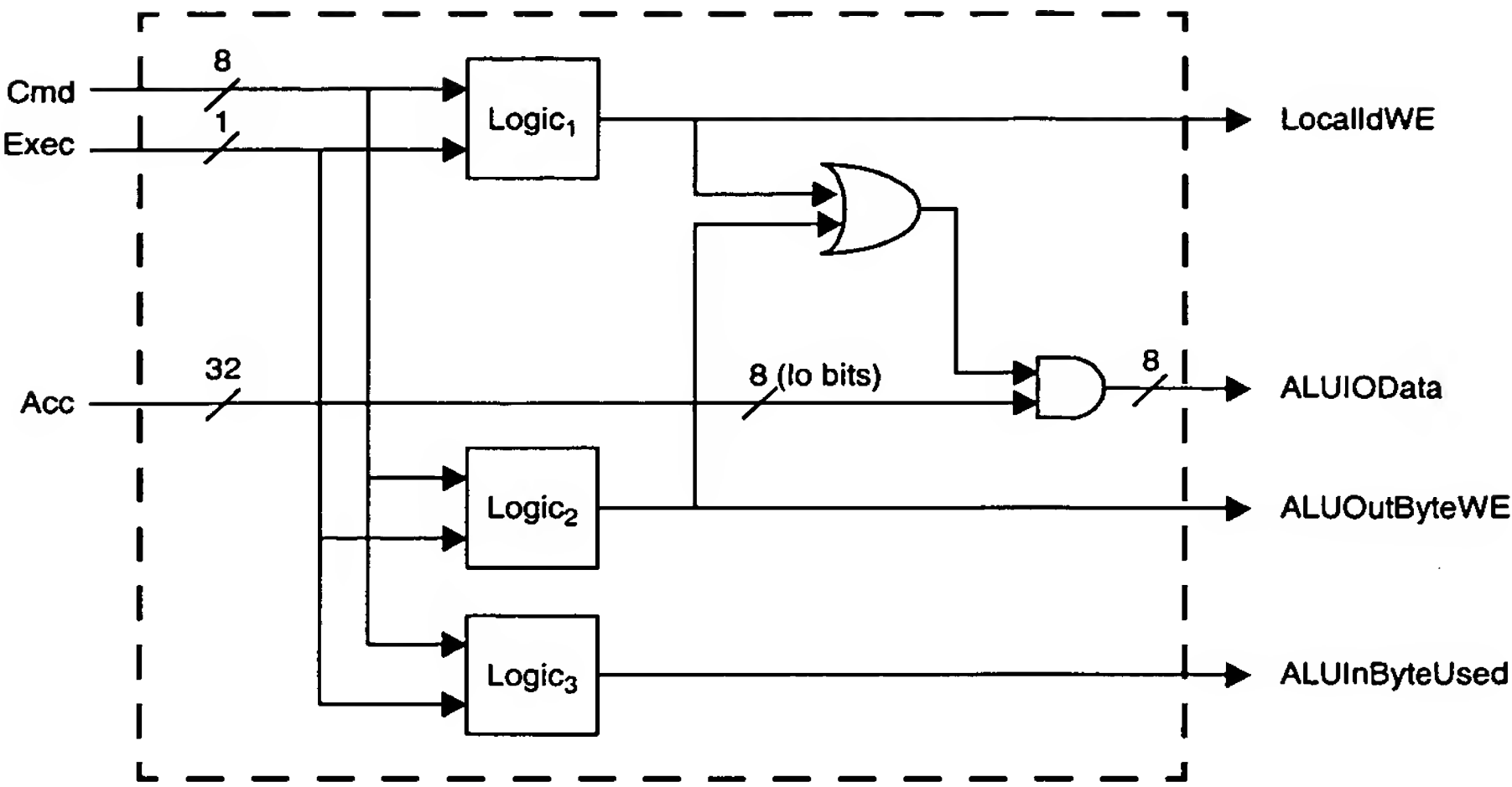


FIG. 402

325/331

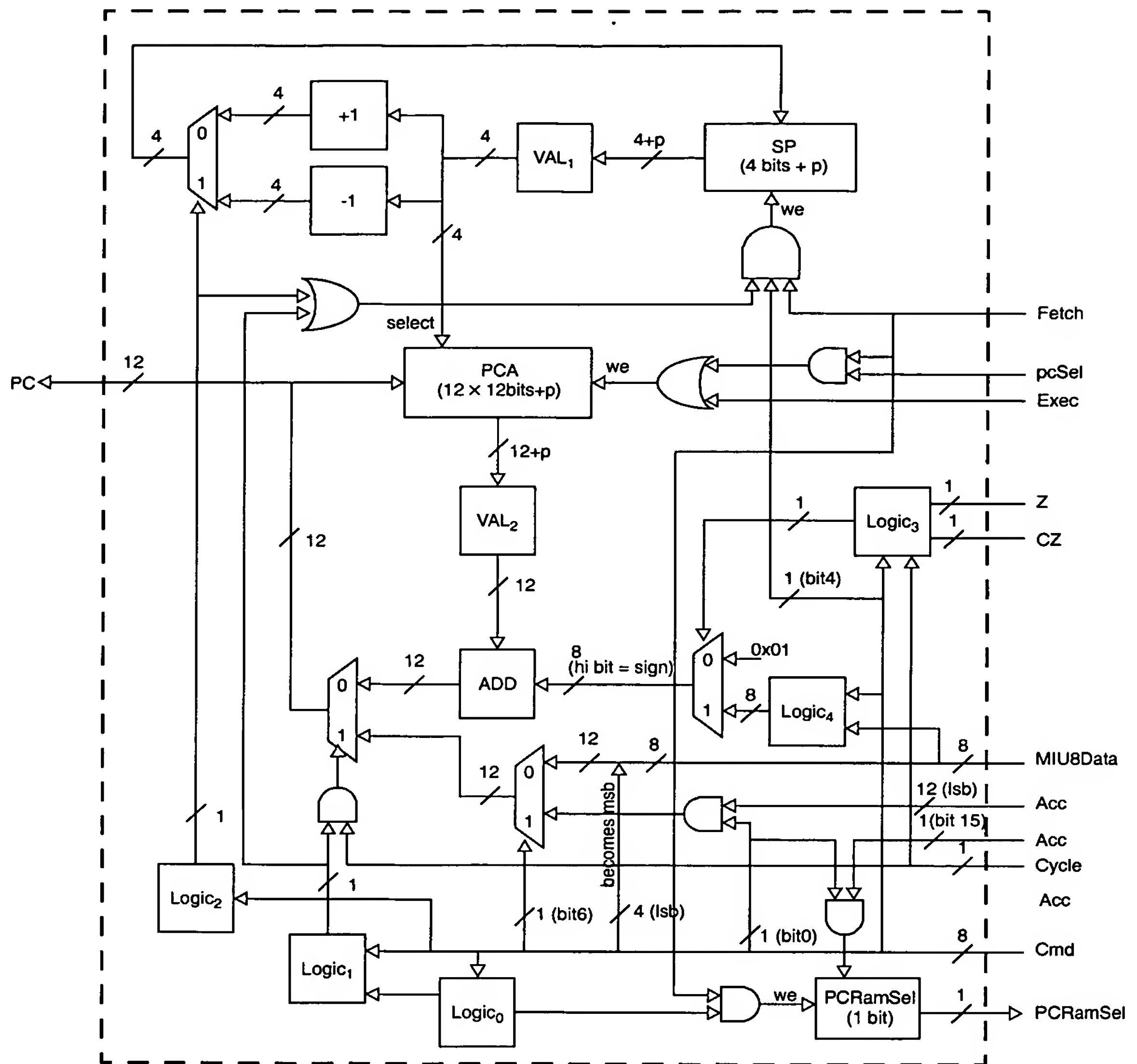


FIG. 403

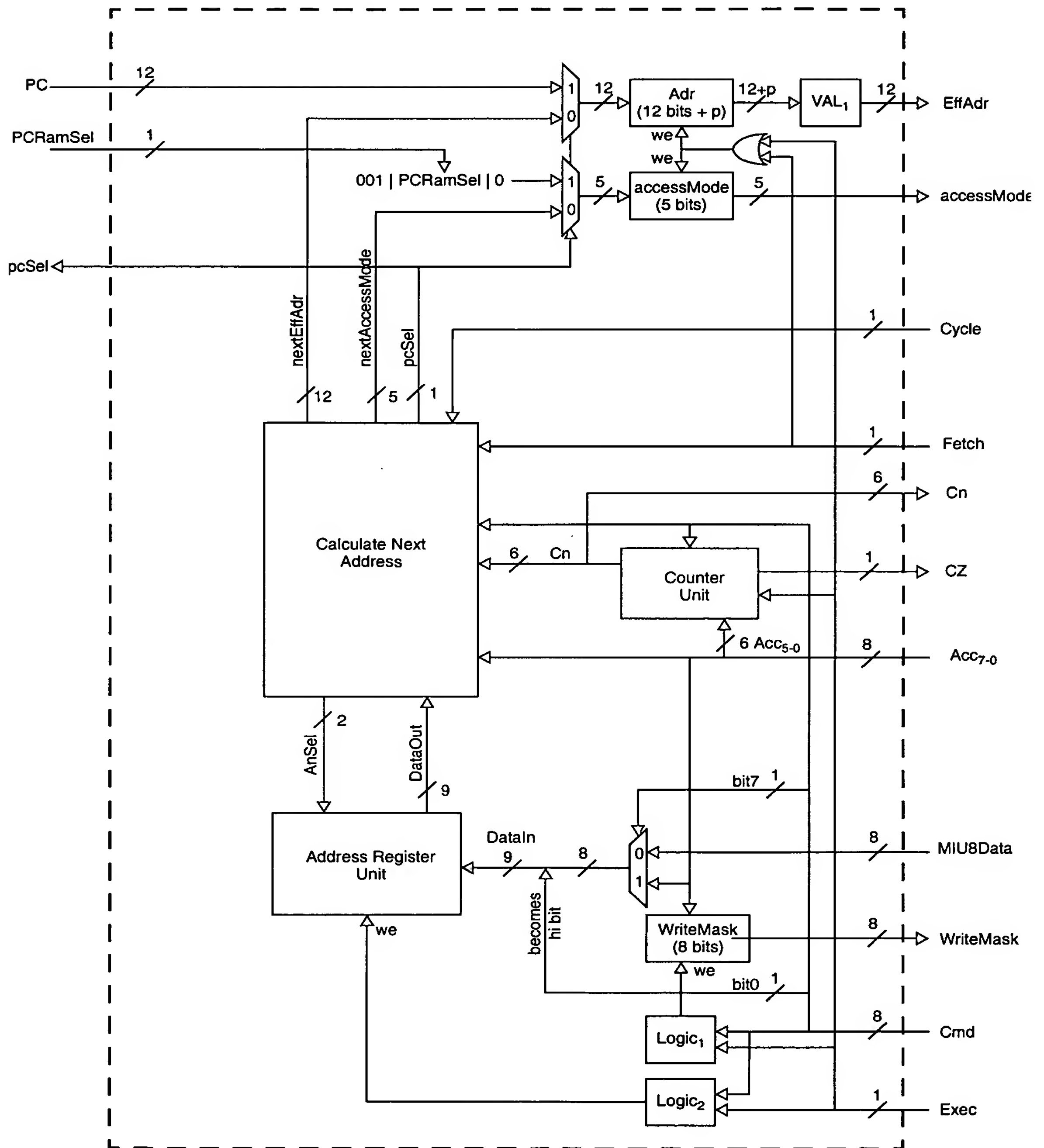


FIG. 404

327/331

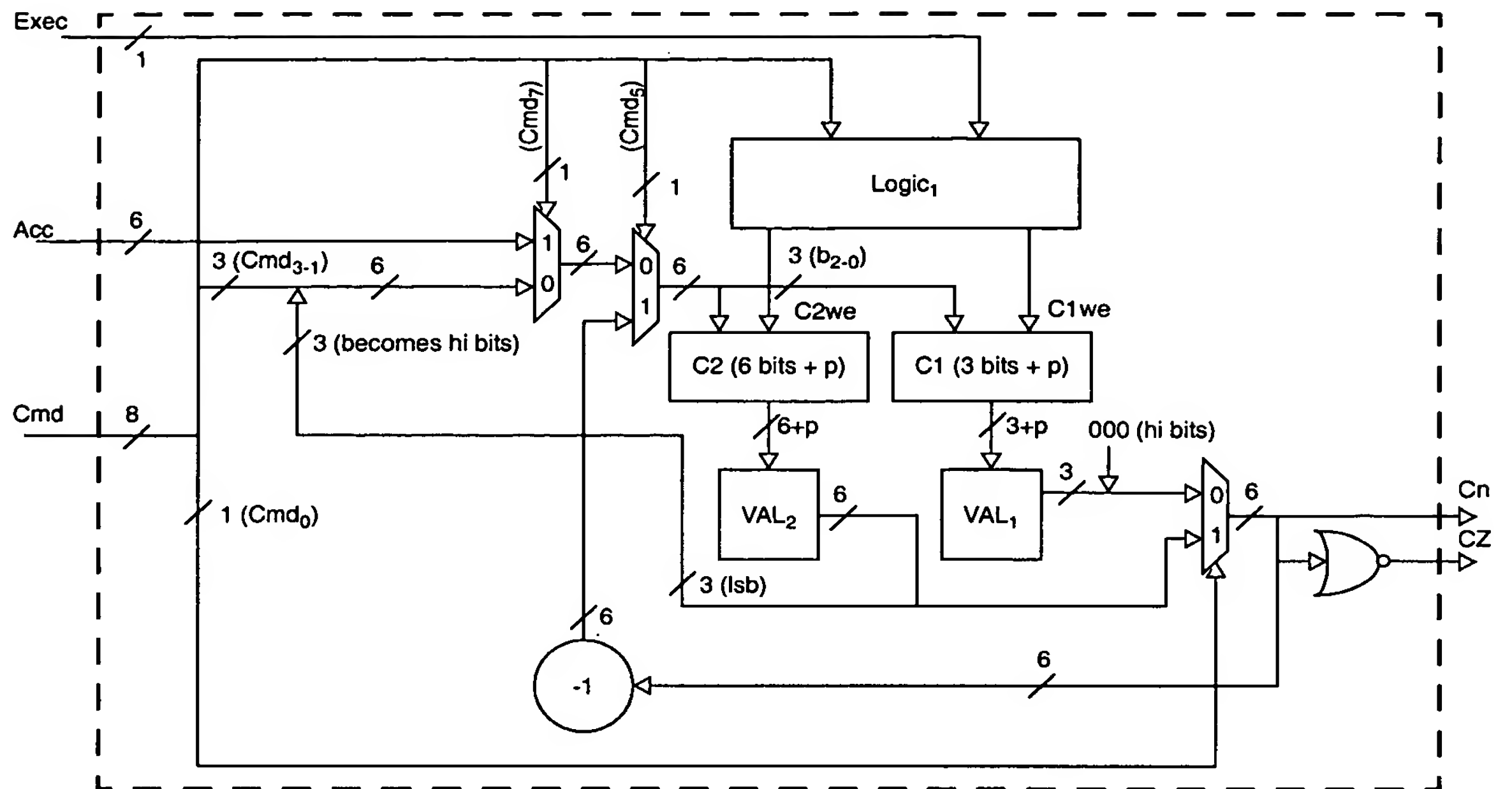


FIG. 405

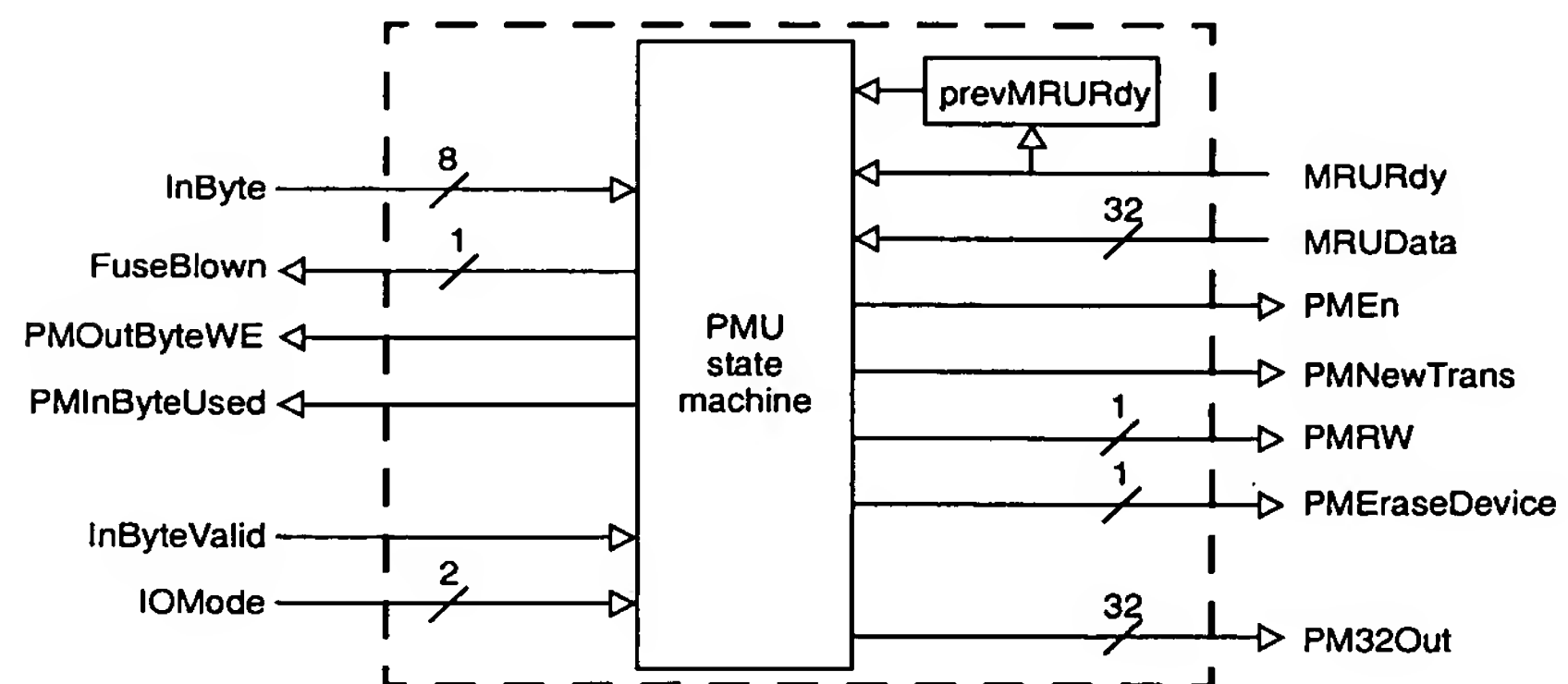


FIG. 406

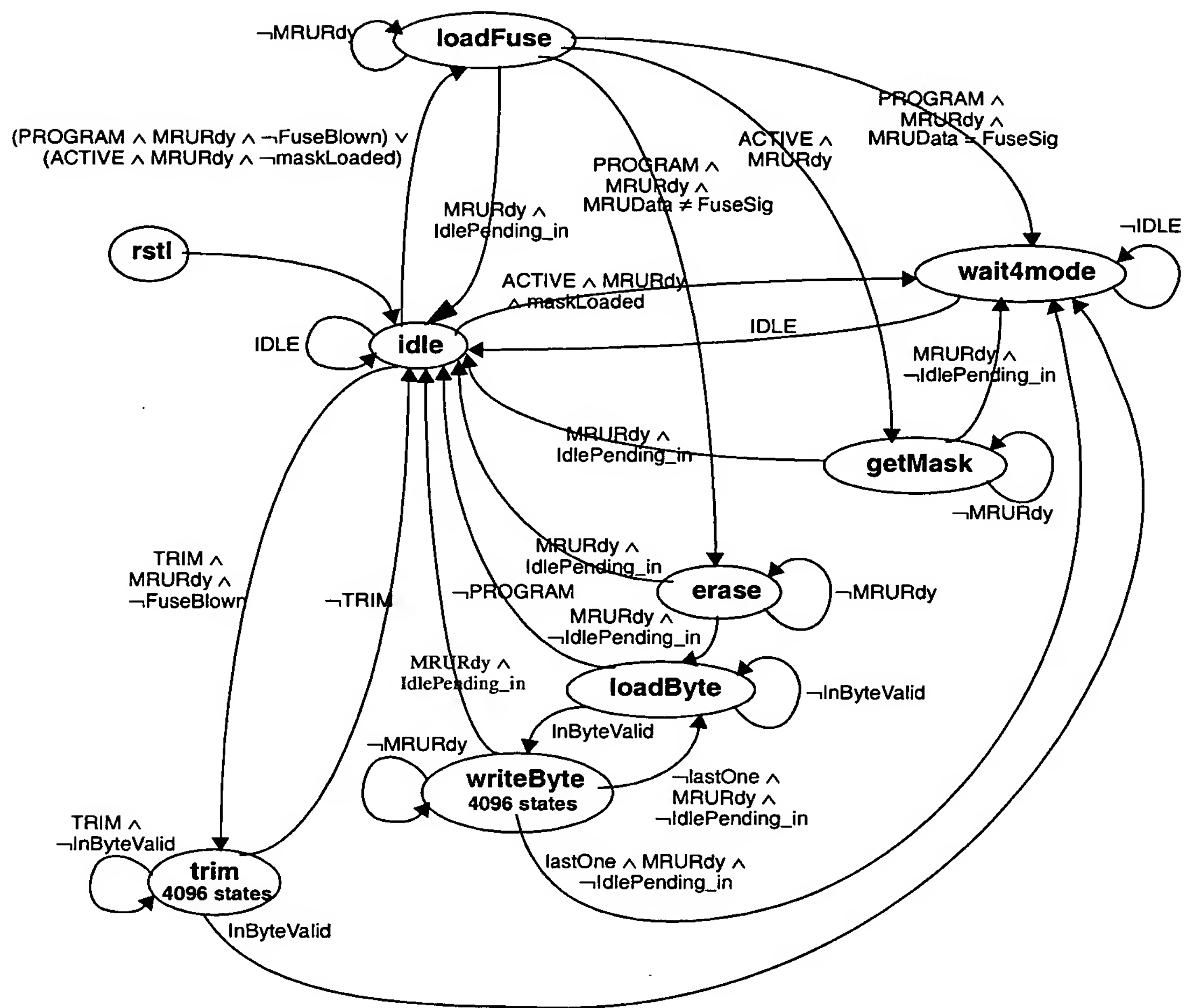


FIG. 407

329/331

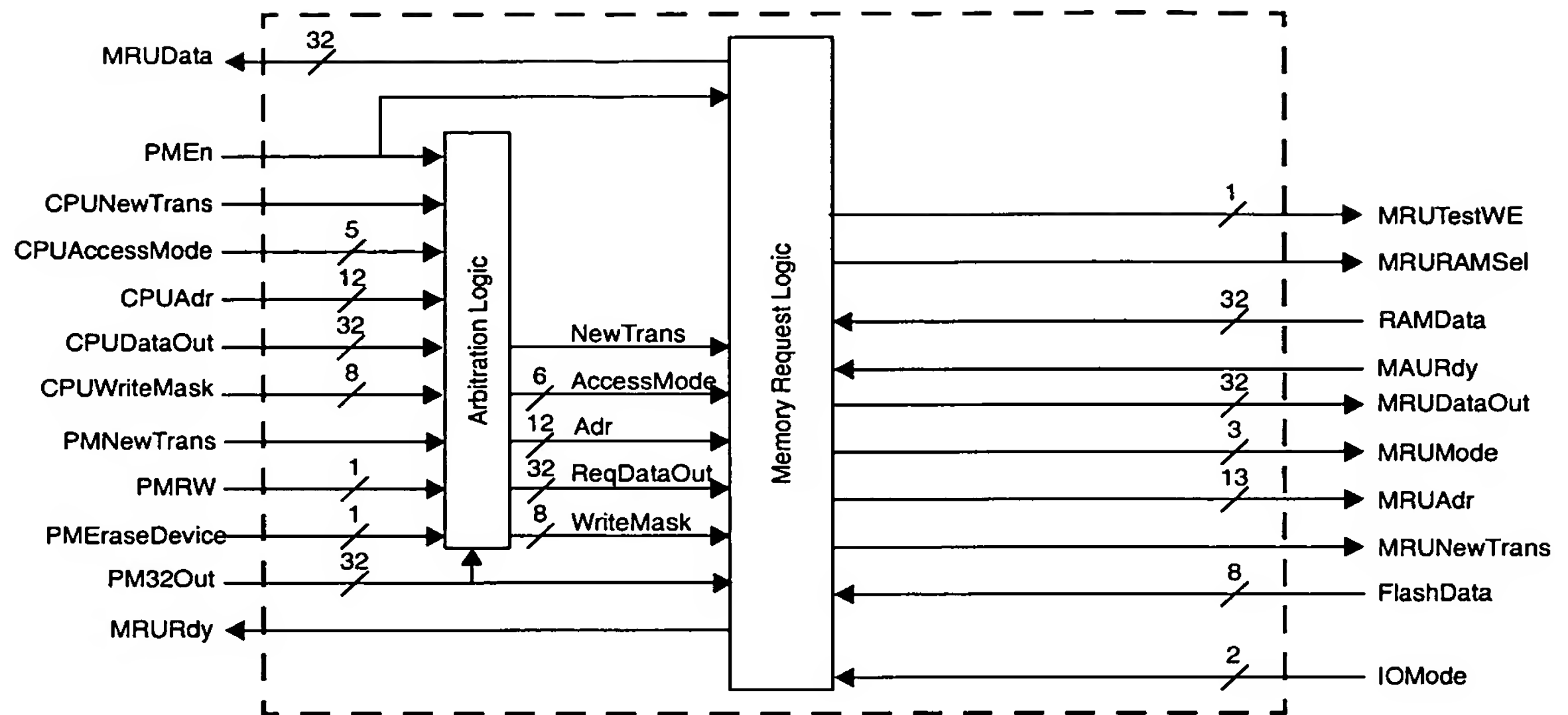


FIG. 408

330/331

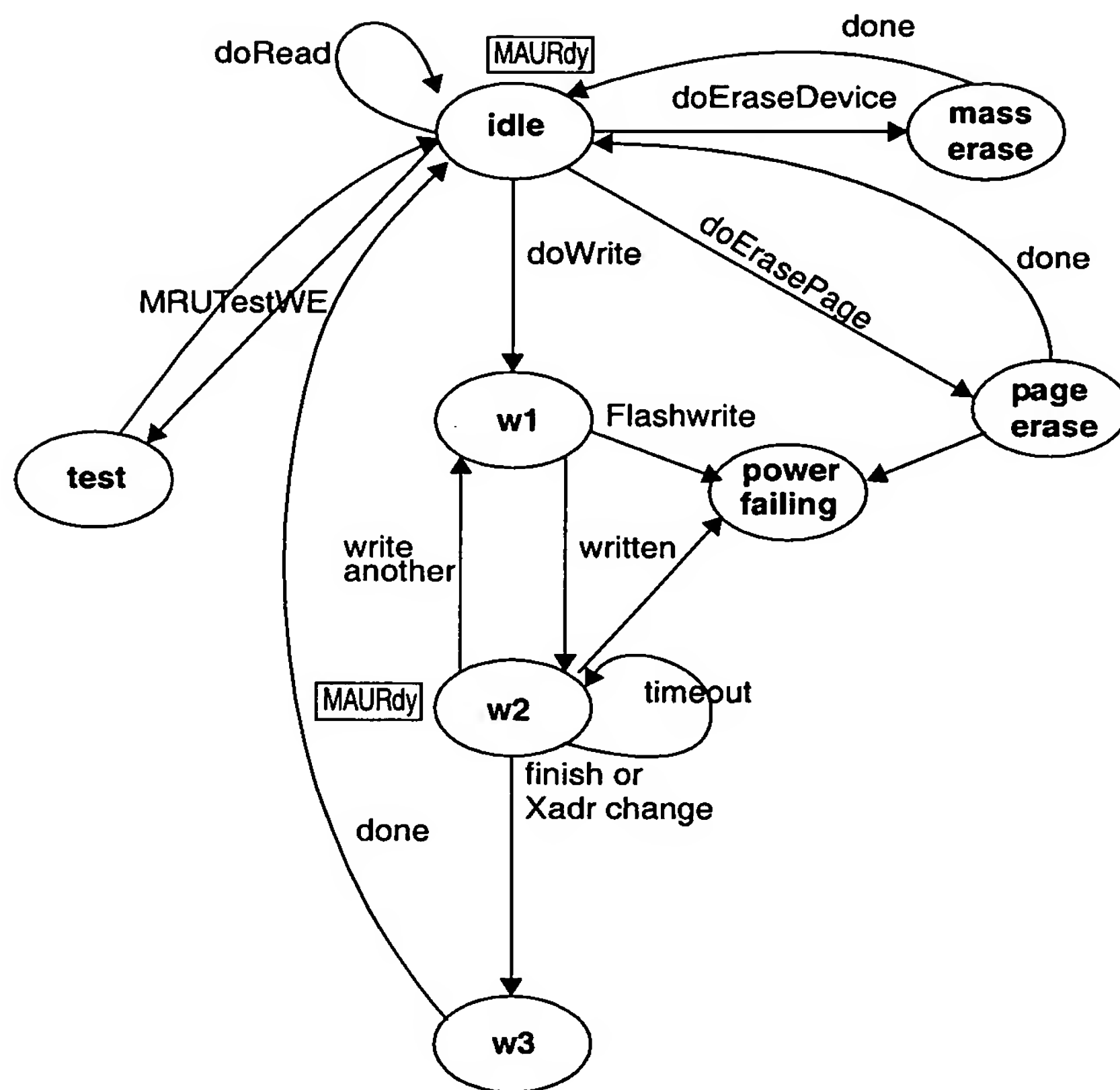


FIG. 409

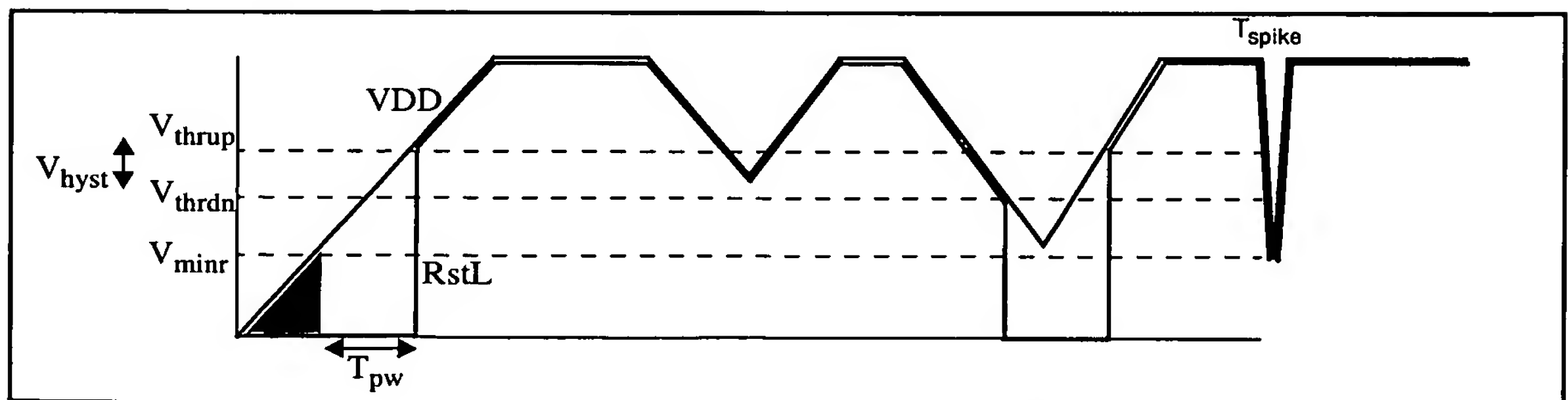


FIG. 410

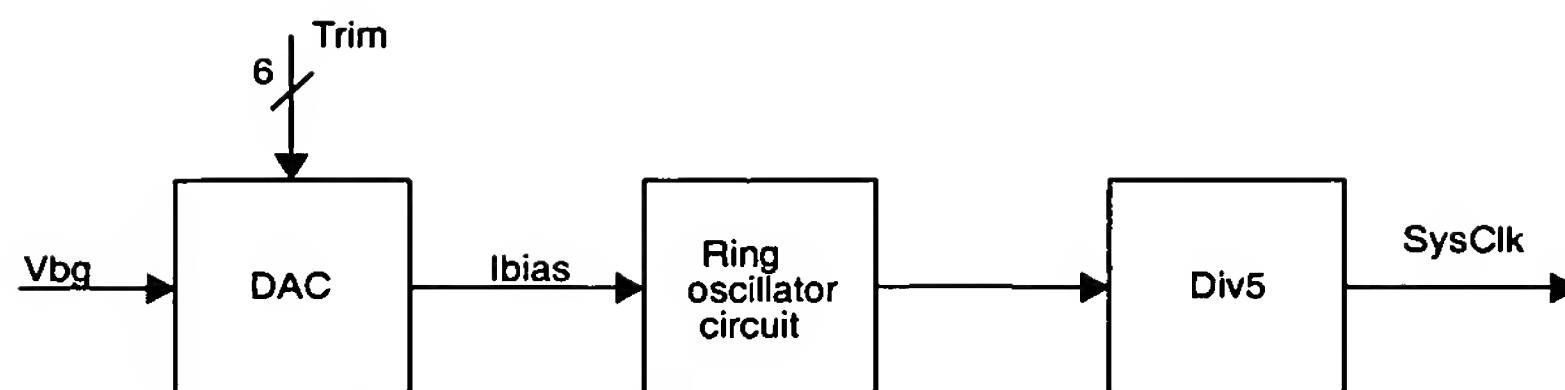


FIG. 411

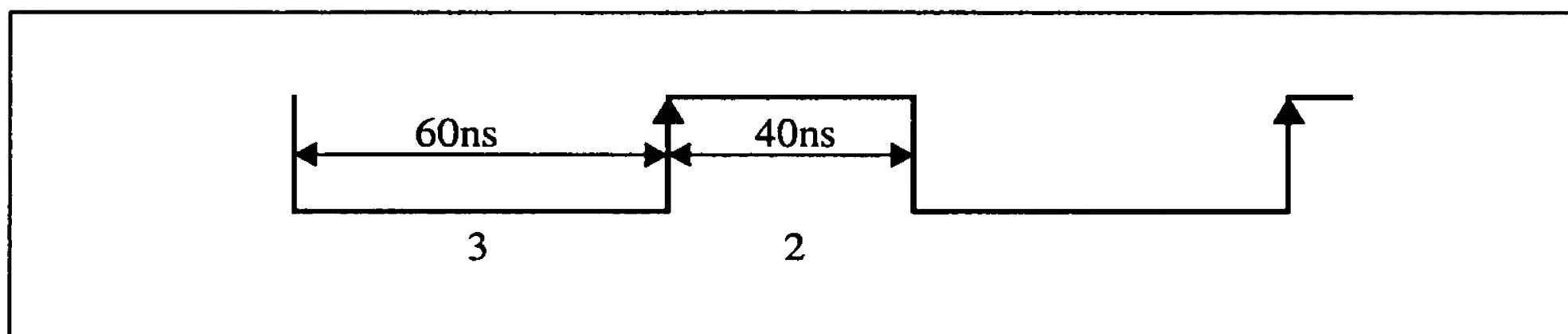


FIG. 412

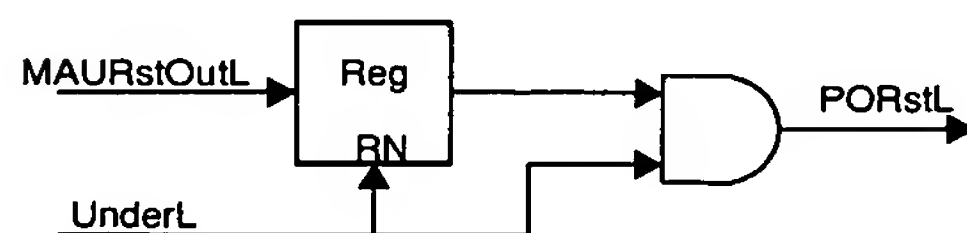


FIG. 413